Geometry Model for Marker-Based Localisation

Alexandre Bousaid
University of Salford

This dissertation is submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Computing, Science & Engineering

Autonomous Systems and Robotics

March 2019

Abstract

This work presents a novel approach for position estimation from monocular vision. It has been shown that vision systems have great capability in reaching precise and accurate measurements and are becoming the state-of-the-art in navigation. Navigation systems have only been integrated in commercial mobile robots since the early 2000s, and yet localisation in a dynamic environment that form the main building block of navigation, has no truly elegant solution. Solutions are many and their strategies and methods differ depending on the application. For the lack of a single accurate procedure, methods are combined which make use of different sensors fusion. This thesis focus on the use of monocular vision sensor to develop an accurate Marker-Based positioning system that can be used in various applications in outdoor, in agriculture for example, and in other indoor applications. Many contributions arouse here in this context. A main contribution is in perspective distortion correction in which distortions are modeled in all its forms with correction process. This is essential when dealing with measurements and shapes in images. Because of the lack of robustness in depth sensing using monocular vision-based system, a second contribution is in the novel spherical marker-based approach position captured, which is designed and developed within the concept of relative pose estimation. In this Model-Based position estimation, relative position can be extracted instantaneously without the need of prior knowledge of the previous state of the camera, as it relies on monocular image. This model can as well compensate for the lack of knowledge in the scale of the real world, for example in the case of Monocular Visual Simultaneous Localisation and Mapping (VSLAM). In addition to these contributions, some experimental and simulation evidence presented in this work has shown feasibility of the reading measurements like distance capture and relative pose between the marker-based model and the observer, with reliability and high accuracy. The system has shown the ability to track accurately the object at a farthest possible position from low resolution digital images and from a single viewpoint. While the main application field targeted is tracking mobile-robots, other applications can profit from this concept like motion capture and application related to the field of topography.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Samia-Nefti Meziani for giving me the chance to be part of a great European project over the last three years. I would like to thank as well Dr. Theo Theodoridis for his assistance, trust and support especially during the final stages of the PhD.

My deepest appreciation to the prompt support of my close friends Stefania, Roy and Lisa to whom I am really grateful; they all helped in many occasions and challenging moments.

Last but not the least, I would like thank my family: my parents and my brother who supported me, believed in me and encouraged me in all of my pursuits throughout the final years and my life, in general.

Preface

The research on this topic was pursued jointly at the School of Computing at the University of Salford Manchester, UK and AGCO/Fendt R&AE Department, Germany, who offered logistics facilities and work environment. The research project has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA Grant Agreement Number 608022.

Mobile Agricultural Robot Swarm (MARS) project addresses looming challenges in real-life to deliver an accurate, robust and low cost multi-robot navigation concept for swarm robots fleet. Part of this research involves designing a system that is capable of overcoming the challenges of localization. The idea behind the project is illustrated in Figure 1.

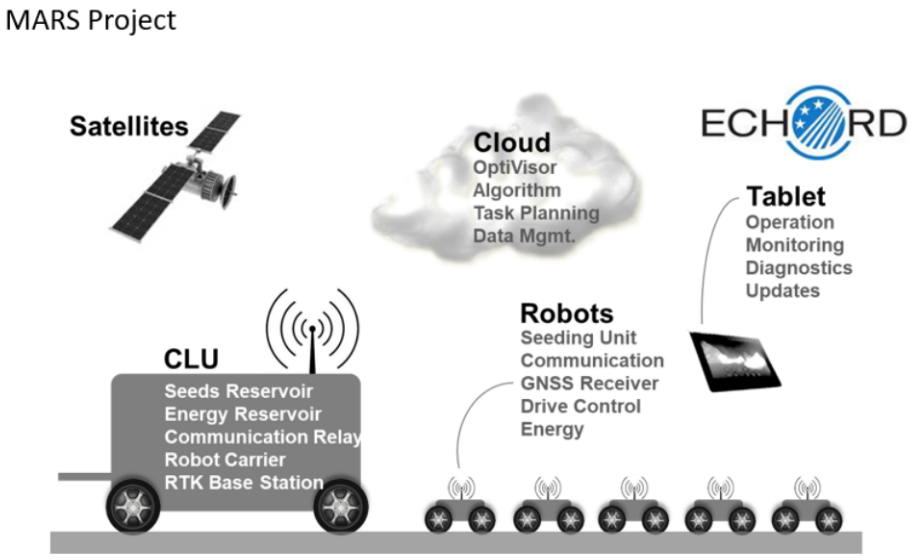


Figure 1: MARS Project. The shown concept was designed from precision farming application that requires accurate localisation.

Contents

List of Figures

List of Tables

Abbreviations and Notations

	Abbreviations
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AR	AUGMENTED REALITY
CBP	CIRCULAR-BENCHMARK PLATFORM
CG	COMPUTER GRAPHICS
COG	CENTER OF GRAVITY
COP	CENTER OF PROJECTION
CP	CENTER OF PERSPECTIVE
CPM	CAMERA PINHOLE MODEL
CT	CIRCULAR TARGET
CV	COMPUTER VISION
DC	DISTORTION CORRECTION
DDF	DISTANCE DISTORTION FACTOR
DF	DISTANCE FACTOR
DG	DESCRIPTIVE GEOMETRY
DDF	DISTANCE DISTORTION FACTOR
DGPS	Differential GPS
DL	DEEP LEARNING
DNN	DEEP NEURAL NETWORKS
EMF	ELECTROMAGNETIC FORCE
FD	FORESHORTENING DISTORTION

FDF	FORESHORTENING DISTORTION FACTOR
FE	FORESHORTENING EFFECT
FOV	FIELD OF VIEW
GBP	GRID-BENCHMARK PLATFORM
GPS	GLOBAL POSITIONING SYSTEM
GNSS	GLOBAL NAVIGATION SATELLITES SYSTEMS
GIS	GEOGRAPHIC INFORMATION SYSTEMS
GD	GEOMETRIC DISTORTION
GOE	GEOMETRY OF ECCENTRICITY
GUI	GRAPHICAL USER INTERFACE
IA	IMAGE AREA
IEE	IDENTIFICATION OF ECCENTRICITY ERROR
KF	KALMAN FILTER
LBP	LINEAR-BENCHMARK PLATFORM
LiDAR	LIGHT DETECTION AND RANGING
MAPR	MID-ARC PROJECTIVE ROTATION
MBO	MARKER-BASED OBJECT
MSE	MEAN SQUARE ERROR
MBV	MARKER-BASED VISION
ML	MACHINE LEARNING
MV	MONOCULAR VISION
MVO	MONOCULAR VISUAL ODOMETRY
OD	OPTICAL DISTORTION
OH	OBJECT HEIGHT
OL	OPTICAL LENSES
PC	PROGRAMABLE COMPUTER
PD	PERSPECTIVE DISTORTION
PDF	POSITION DISTORTION FACTOR
PE	PERSPECTIVE ECCENTRICITY
PE	POSITION ESTIMATION

PE	POSE ESTIMATION
PE	PERSPECTIVE EFFECT
PF	POSE FACTOR
PD	Perspective Distortion
PDF	Pose Distortion Factor
PG	PERSPECTIVE GEOMETRY
PG	PROJECTIVE GEOMETRY
PnP	PERSPECTIVE-N-POINT
PP	PROJECTIVE PROJECTION
PP	PARALLEL PROJECTION
PP	PERSPECTIVE PROJECTION
PR	PROJECTIVE ROTATION
RADAR	RADIO DETECTION AND RANGING
RTK	REAL-TIME KINEMATIC
SA	SOFTWARE ARCHITECTURE
SP	SPATIAL PERCEPTION
SLAM	SIMULTANEOUS LOCALIZATION AND MAPPING
SONAR	SOUND NAVIGATION RANGING
SR	SPACE RESECTION
ST	SPHERICAL TARGET
T	TRANSFORMATION
TCL	TARGET CENTER LOCATION
TDF	TOTAL DISTORTION FACTOR
V-SLAM	VISUAL SIMULTANEOUS LOCALIZATION AND MAPPING
VCS	VIRTUAL CIRCUMSCRIBED SPHERE
VO	VISUAL ODOMETRY
VR	VIRTUAL REALITY
VS	VIRTUAL SPHERE
VS2	VIRTUAL SPHERE 2

WF	WORKFLOW

Chapter 1

Introduction

Contents

1 Motivation

This thesis is intended to investigate into the development of a perspective position estimation from monocular view. Although the focus is on position estimation, some part of the work is also dedicated to investigating the recovery of the full pose from monocular view. Pose estimation refers to the problem of determining the position and orientation of a camera center relative to a targeted object in the scene or vice versa. Pose estimation problem addresses looming challenges in real-world localisation, constrained by environmental, economical and technical aspects. The ability to localise mobile objects in indoor or outdoor space remains substantially challenging (141).

Localisation forms the main building block of navigation (21). Yet, there are no truly elegant solutions (138), creating a bottleneck on navigation systems by preventing seamless and accurate positioning in different environments. While huge loads of application require high precision positioning, this leaves the door open to ongoing research on this topic for improving the performance (141).

Many systems nowadays rely on rails, guided wires or wireless radio technologies in environments with tendency to Electromagnetic Field (EMF) interference such as warehouse, hospitals and related indoor spaces (136), (110) (79). In addition, Global Positioning System (GPS) technology (99, 82) is not accessible in buildings and is restricted to positioning in outdoor areas, where satellites bind images cover most areas for creating 3D maps. GPS lack reliability in changing and harsh environment and are inaccurate for navigation (12). In contrast for example, Real-Time Kinematics (RTK) are accurate differential - Global Navigation Satellite System (Differential-GNSS) (72). RTK techniques are based on measurements on the carrier signal performed at the based station at a well known position, which then transmit a correction signal to the GPS receivers (68). This makes the RTK extremely expensive, and require setting up base station on the fields, which demand huge space and maintenance cost (105). With regard to the advantages of these technologies, their disadvantages can arouse limitation to their application, and provoke decrease in production and productivity, and can lead to time and resources consuming.

Vision systems in the context of localisation, have their own specificity. Broadly speaking, there exist many types of vision systems, from 1D and line scan, 2D and Area Scans to 3D systems (92). Vision sensors have wide variety of applications, specifically in mobile robotics (31). A lot of interest on such sensor nowadays is taking part in autonomous transport. The two recent most debated and researched are Light Detection and Ranging (LiDAR) and cameras. LiDAR (71) is a light-based Radio Detection and Ranging (RADAR) (35), (64), a remote sensing technology used to measure distances. LiDAR uses the same techniques as the sound navigation ranging (SONAR), it bounces a probe beam off to the target and generates distances based on the propagated time of the echo (1). Using the range plotting, a map of the environment can be drawn. A major advantage of LiDAR is that it works independently from the ambient light (35). It sees the same under any condition, whether it is subject to daylight or night illumination, cloudy or sunny conditions. Another advantage of LiDAR on RADAR is that it offers the capability of constructing a $3D$ map of the environment around at much higher image resolution, in addition to its immunity to interference (1), (54). However, the range of LiDAR, when fused with other sensors, can vary from 150m to 700m with accuracy, starting from 25mm, depending on the conditions and application (54). As compared to camera sensor, LiDAR is very expensive, have lower resolution but can be much accurate when integrated with multi-sensors (54), and have much slower refresh rate, which make them inaccurate in scanning moving objects or while moving the device itself (24).

Like other vision sensors, cameras rely on physical stimulus which is light. Camera sensors use natural lights to produce a flipped image of the reflected scene during daylight. Night vision requires the addition of Infrared Red (IR) transmitter in order for the camera sensor to capture images (65).

The camera model is constituted of many separates layers which we will explore more in details throughout this thesis. These layers are illustrated in figure 1.1.

Camera sensors are extremely inexpensive, small in size and users can benefit

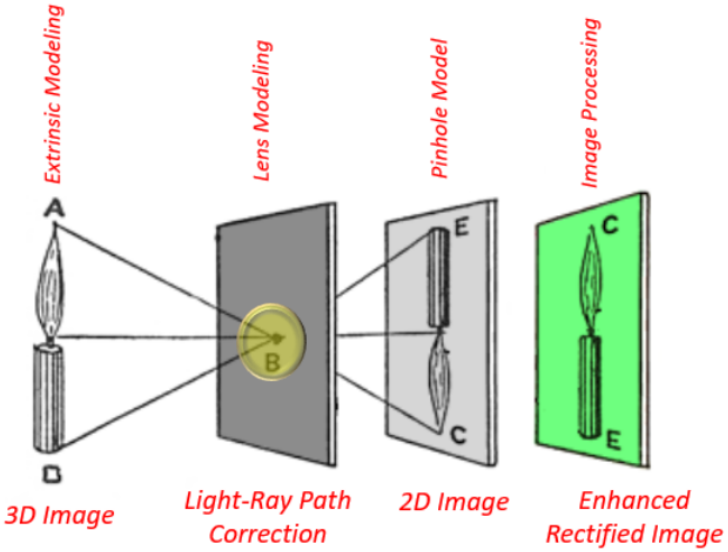


Figure 1.1: Camera Model Layers.

from too many features like multi-usage for the same application, colourful understandable image, 3D reconstruction and high resolution images provided (131). There are two important aspects for vision systems, resolution and sensitivity; the former defines the ability to pick out fine spatial details from the image sensor (86), while the latter, refers to the ability to detect light at different illumination level(86). From these two aspects, one can define the specification of the camera sensor based on the required application.

Localisation problem and pose estimation from camera images have become a fundamental subject and the central task in computer vision and photogrammetry (104), (65). It has taken the attention of scientists and researchers since decades, when it was first referred to as space resection by Moffitt Mikhail in 1980 (118).

We worked in the thesis on the development of a new approach for position estimation with a main motivation to create a robust, inexpensive and green solution. We refer to the camera solution as green since it does not transmit any harmful signal in addition to its supremely low power consumption. We describe the camera sensor by the following attributes:

- it is a low cost and green solution.
- it senses the environment in multi-dimensions including colours.
- it performs multi-task at once
  - it generates a map of the environment
  - performs motion, obstacle detection and tracking
  - performs depth extraction and position estimation
- it is a deaf sensor that cannot sense from all directions, but it can see light at an angle.

From a mathematical point of view, current existing models for pose estimation, like the perspective-n-points also known as *pnp*, require that the position of a given n $3D$ reference points of a target in the camera scene to be known, such that $n \geq 3$ (46). For $n = 3$, the problem is considered to be a $p3p$ with eight different solutions or poses generated. Solving the full pose on the camera requires a number of known points in the real world to be greater than four (66). During the work of this thesis we investigated a position estimation model that could deliver a solution for the 3D position out of three points in space regardless of the orientation. We looked at the problem from a different perspective by splitting the pose into position and orientation and then solving each separately. We came out with a solid mathematical model with proof of concept that allows capturing object shapes and project the relative position of this object out of its geometry.

2 Main Contribution

We present here our main contributions related to the work presented within the scope of monocular vision. We believe these contributions will have impacts on vision-based application related to camera pose estimation, object tracking and recognition and vision based-navigation approaches. On a specific level, we have formulated and modeled perspective distortions from a geometrical aspect, which contribute in providing proofs for understanding to the nature of perspective distortion and open the door to finding ways to distortion elimination.

2.1 Perspective Distortion Modeling

A considerable portion of the contribution of this thesis is on perspective distortion modeling, which we discuss in chapter 3. This contribution plays a big role in classifying the three main distortions of perspective project based on John Y. Aloimonos description in (6). These distortions are:

- Distance Effect
- Pose Effect
- Foreshortening Effect

We model every distortion and provide mathematical proofs followed by simulations and experimentation. We show the effect of all these distortions on different types of targets like line segments, circular flat targets, spherical and triangular markers. We then show how to use these models in position and orientation estimation in 2D and 3D.

2.2 Marker-Based Localisation

Figure 1.2: The first Marker-Based sketch.

A contribution on marker-based localisation is explored in chapter 3. We show in figure 1.2 a sketch of the marker approach usage. We designed a novel geometry projection model depicting the projection of three points in space into a 2D image plane. We created two separate models which capture the parallel projection

of the distances between the three vertices and generate the radius of the circumscribe sphere across the three vertices. We named these models the Virtual Sphere 1 and 2. The VS 1 works for equidistant vertices while the VS 2 represents the general form of projection created by any three points in space. These are the main building blocks of our approach for solving the position estimation from monocular view. We present in chapter 4 all the mathematical proofs along with simulation and experimentation.

2.3 Distance Estimation in Monocular Vision

In chapter 3 and 4 we discuss possible ways for distance estimation to lines and spheres in monocular vision. We back this up with the mathematical model developed and prove that it is possible to improve the resolution of measurements from one side which allow the use of a low cost sensor from the other.

2.4 Position Estimation

In chapter 4 and chapter 5, a big contribution in position estimation from three points in space is explored. We use our previous developed models of corrections to estimate the accurate relative positions, where we show also the derivation of the full pose in some cases.

The results are shown in real life experiments in chapter 5. For the purpose of generating the ground truth, we designed and built three benchmark platforms in order to measure and evaluate all the three distortion effects on the target. These benchmarks are:

- Linear-Benchmark Platform
- Circular-Benchmark Platform
- Grid-Benchmark Platform

2.5 Scale for Feature Based SLAM

A part of the work is dedicated to showing the integration of novel marker geometry in calculating the SLAM scale from monocular vision. The part is explored more in details in chapter 6. We show that this model can as well compensate for the lack of knowledge in the scale of the real world, as the scale of a world cannot be determined using single camera without relying on commissioned real landmark. We present an approach for solving the SLAM Scale by combining the scale factor with the SLAM world pose estimation as in the case of Monocular Visual SLAM.

2.6 Cooperative Localisation

Finally we want here to point out to an approach that could contribute in cooperative localisation for multi-robots navigation. This approach is discussed more in the last chapter 7. We drew the concept in figure 1.3.

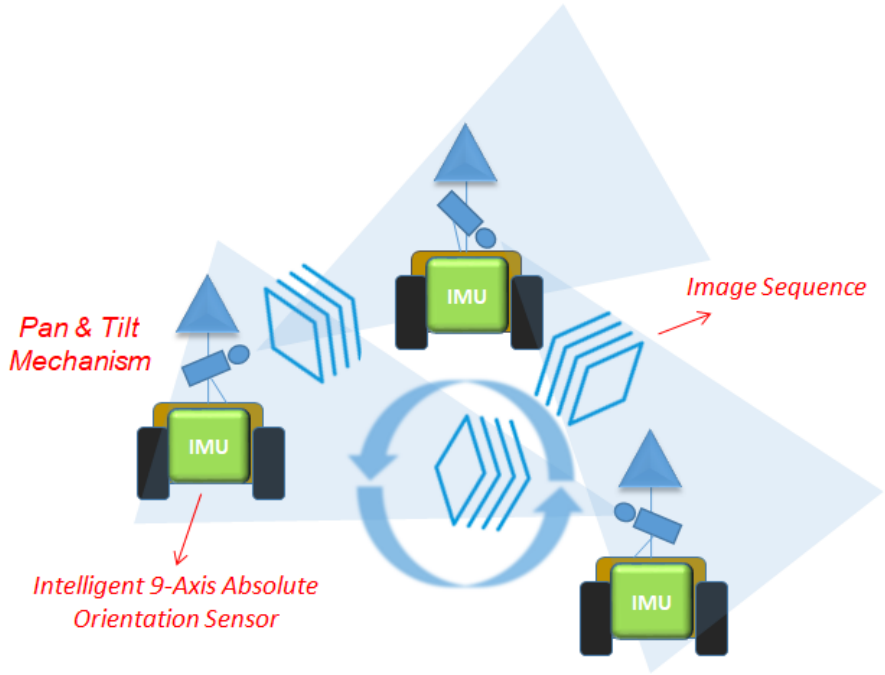


Figure 1.3: Marker-Based Cooperative Localisation Concept.

We show here that with the use of IMU, and other sensors fusion while extracting

features from image sequence to keep track of the individual position, one can benefit from a great ability to accurately localise a fleet of robots.

3 Thesis Outline

In this work, several models were developed in this regard for solving pose estimation, where we use a main target object consisting of three points in space, from which the relative pose is the subject of study. We have highlighted in the earlier sections the purpose and scope of the thesis, we will now explore the contents of every chapter.

In Chapter 2, we provide introductory concepts and literature on image based localisation techniques which are related to vision systems. We followed by outlining image formation and by giving projective geometry background with focus on perspective projection, which is the basis on which image is formed by camera sensors. In later sections we present background and functionality of a pinhole model adding to it calibration model, to finish with recall on 2D and 3D transformation in homogeneous coordinate system.

In Chapter 3, we explore a unique distortion modeling for perspective view and use these models to demonstrate distance and position extracted from different targets, by means of mathematical modeling. We then present correction methods using projective rotation methods. We backup these models by providing simulation evidence on spherical targets in images, where we show the importance of correcting spherical eccentricity in images. We finish with summary and discussions around the results.

In chapter 4, we present two projection models using parallel projection of 3D shapes of 3 points into a 2D plane. In the first model we consider that the object or target in the scene is of specific triangular geometry whereas the second model represents a general formula of the projection that fits to any particular target shape and from which position and orientation are extracted. These models are referred to by VS1 and VS2. We then show how it is possible to recover the

position from monocular view of such targets.

The experimental and simulation evidence are presented in chapter 5 for further proof of the models. We designed three test benchmarks specified to every case of distortion: One that captures distance effect, a second one that is designated to evaluate pose effect and a third one which is allocated to evaluate foreshortening effects in random motions. We also show in these experiments the ability to track accurately the object at a farthest possible position from low resolution digital images and from a single viewpoint.

In chapter 6 we explore our contribution in calculating the scale which is a necessity for SLAM-Based systems that use monocular vision. We present our approach which relies on geometric constraints to solve the scale drift in SLAM. We show by experimenting and integrating the approach into the ORB-SLAM open-source SLAM, that we were able to restore the scale efficiently without the need to commission the environment.

Finally, in chapter 7 we close with a conclusion on the findings and limitation of the work and propose future work based on the current achievements.

Chapter 2

Background

Contents

1 Chapter Summary

In this chapter a review of the literature is presented with technical backgrounds, which introduces concepts that are relevant to the work confronted.

We start by covering the state-of-the-art of image-based camera localisation. We then discuss more in depth existing position estimation strategies and techniques. Additionally, we include an overview on marker-based techniques and monocular SLAM.

The-state-of-art of image formation is then presented, where we explore projective geometry focusing on perspective branch. We then relate the theory to image formation in addition to outlining camera pinhole model.

2 Image-Based Camera Localisation

Camera sensors can capture images at high frame rate and can be extremely inexpensive, which make them most of the times more favorable for many applications (135). With camera data, a large amount of information can be extracted (85). This has led to intense investigations on vision-based system for mobile robotics that started early 1980s according to the survey presented in (34).

In the case of mobile robotics, vision system can be dedicated to work either indoor or outdoor (34), for many reasons that we list here as follow. The first reason is that features in indoor can be different than outdoor features in images, in terms of illumination, sizes and distances. Second, light changing conditions vary in much larger range and frequency. A third reason is that outdoor environments are considered to be much more dynamic and are much larger for storing such big amount of features and landmarks information. This also leads to classifying vision systems into structured or unstructured environment. In structured environments, dense sets of points or landmarks are present in order for the system to localise itself (70), whereas, in unstructured environment, the

system must extract possible features from the scene (70) to achieve localisation. Many approaches are addressed in this regard in (120), (125). Other approaches falling under environmental structure classification are whether systems are map-based or map-less (40).

Accurate localisation can be significantly improved in map-based systems by using a built-in map of the area of operation. According to (40), there are many algorithms that require computed map beforehand, like multi-state constraint kalman filter, parallel tracking and mapping (PTAM) (97). Map-based techniques can also be subdivided into Metric-Mapping, Map-building and Topological Map-Based according to the survey on vision-based localisation (12). Metric-Mapping needs to be loaded with map beforehand and topological Mapping uses graphical representation consisting of linked nodes (132), (117), while Map-Building localisation needs to explore first and create the map in real-time. The latter is the most famous technique nowadays, also known as Simultaneous localisation and Mapping (SLAM) and Visual Simultaneous localisation and Mapping (V-SLAM) (115), (18), (36), (76), (63), depicting visual systems. A general overview of the SLAM problem is detailed in (39) and (9). These techniques are discussed later in section 2.3.

On the other hand, map-less strategies achieve localisation without any prior description of the environment, but rather by observing features of this environment (12). It does not require building a map. Map-less localisation can also be divided into two sub-groups which are Optical-Flow (OF) and Feature Tracking (FT), according to (12) and (14). Finally, localisation in general has to be classified accordingly to references in space and time. These two classifications are absolute localisation or relative localisation, known as well by global and incremental localisation, respectively (116). Absolute vision-based localisation does not require any prior position knowledge and can be localised at every instance. Conversely, relative localisation can be extracted relative to prior knowledge from stored image database (143), (116).

2.1 Monocular Vision

Vision-Based systems can also be classified based on sensor hardware configuration groups(12). Mainly the largest part of application uses monocular sensors. Other hardware types that are very useful for 3D reconstruction, mapping and localisation are based on binocular sensors. These systems are grouped under the name of stereo-based systems (96). Further groups exist which are trinocular, which use three image sensors, and omnidirectional sensor (137), (22), which uses hyperbolic mirror or omnidirectional mirrors that are placed on top of the camera sensor. Omnidirectional mirrors allow light coming from all direction to be reflected into the image sensor, allowing wide angle of view up to 360 degrees images.

The importance of monocular vision has been stressed in many surveys (26), (93) and publications (83), (67). Many applications rely on such configuration like real time localisation (105), video surveillance, and various robotics applications including V-SLAM. With monocular sensors, depth cannot be fully recovered. The motion parameter is provided up to a scale factor which is the result of perspective projection from 3D to 2D (12). We list next many solutions.

Recovering the full scale is a primary problem in Monocular Visual Odometry (MVO). Image measurements are evaluated up to a scale factor because of the loss of depth in monocular vision. In MVO, the motion parameter cannot be estimated accurately but up to scale factor(119). Because the camera is in motion, the absolute scale factor cannot be estimated and therefore tends to drift due to the accumulation of error (119). A technique to recover the scale factor is introduced in (139). The authors provide a method to incorporate depth estimation from images using deep convolutional neural fields. They show scale recovery from scale drift using structural image data (139). In (29), the authors use texture analysis classifier to extract the ground plane, then apply homography transformation to recover for the scale. In (73) the authors use self-learned ground appearance information to estimate the ground geometry, which make the solution robust in different scenes. Other techniques use geometry constraints between the camera and the surroundings (29) like, for example, height of the

camera, (123) or by using constraints on the kinematic model to estimate the ground truth with the help of odometry data (73). However, these approaches using initial measurements, and assumptions that can face scale drift problem as a result of accumulated error over time (69). This error accumulation comes in turn, from the fact that scale estimation is strongly dependent of the previous estimate.

In summary, we showed previously some solution for scale drift relying on image data; some of these methods are constrained by limited environmental information making their application too specific (66). Many other approaches try to solve the scale problem by integrating more sensors such as RADAR, odometry and IMUs, or for example, by computing the ego motion at the cost of accuracy and computation power (28).

Later in this section, we will explore our strategy and method for solving scale and absolute positioning by introducing a geometric constraint, provided by a novel marker-based approach.

2.2 Marker-Based and Marker-Less

A marker is such a sign that computer systems can detect from image sequence using image processing, pattern recognition and image processing techniques (84). Once the marker is detected, correct scale and pose can be defined. Marker-based Vision (MBV) systems are designed to recognize fiducial markers (37) or targets in two different localisation approaches (109). One approach which is used for motion capture and tracking in which markers are placed on mobile objects (16), (44), (2), that can be tracked using vision system. A second approach is to use the other way around, where the vision system relies on fixed passive or active marker (136) or landmarks to navigate and localise itself within the environment (101). In both approaches relative pose of the camera with respect to the marker has to be extracted. Many low cost autonomous navigation system are based on fiducial marker as stated in (91), (101). On the other hand MBV has various applications in Augmented Reality (AR) (61), programming by demonstration, and imitation of humans in a flexible manner, using passive markers. The popularity

of marker-based systems is given by the fact that they can be easily implemented and because good and well-known marker-based toolkits are available, like ARToolKit (25), ALVAR, ARTag (45).

As opposed to marker-based, marker-less vision systems are based on unconstrained images, which is quite challenging. With unconstrained images, no prior assertions are made to scene (91). Predefined structures and dynamics in the scene are used in order to achieve reliable tracking (5). Intense image processing is employed for the purpose of gathering structural properties of such environments. This big amount of processing, for handling complex real-time tasks, is at the cost of expensive dedicated hardware (144). These approaches are very challenging when it comes to realization due to their complexity and computational demanding algorithms (91), (144).

Marker-based localisation has proven to overcome challenges related to the real world scale estimation and pose estimation drift correction. This is also known by Marker-based tracking approach of simultaneous correct scale and pose estimation. Furthermore, famous systems like the Vicon vision system (88) use markers to track and position systems, using n views from n number of cameras. This system is employed in (38) for flying autonomous quadrotor with four spherical markers mounted. In other applications, this system is used at MIT (60) and Boeing (13) for autonomous indoor flight simulations and demonstrations. This shows the importance and ability of tracking with markers in order to reach accuracy below centimeters.

2.3 Simultaneous Localisation and Mapping

SLAM was first initiated by Smith and Cheeseman (122) in 1986, and became popular in early 1990s (19). Several approaches for solving the localisation problem using vision have been proposed (26). SLAM is an attractive solution since localisation can be achieved without prior knowledge of the environment (129), (36). As soon as the system starts, localisation and map-building happen simultaneously (105). According to survey (129), the SLAM technique was first

proposed to achieve autonomous control of mobile robotics (27). Since then, SLAM started to become widely used in 3D environment modeling applications, AR and most importantly, autonomous navigation (19). Initially the SLAM algorithm has been employed with various types of sensors such as wheel encoder, laser scanner, inertial measurement units, GPS and cameras (7), (39).

Camera based SLAM techniques have simple configuration and are known by V-SLAM. They are widely used in computer vision and AR (129), (53). Although V-SLAM is considered to be heavy on hand-held devices, various low computational V-SLAM algorithms have been proposed. These are summarized in (129). Many technical difficulties arise from V-SLAM compared to other sensor-based SLAM. The limitation comes from the fact that cameras have narrower field of view.

V-SLAM can be implemented using most of the sensor hardware like monocular, stereo, RGB-D and omnidirectional (23). Monocular V-SLAM has become very popular since it relies only on a standard camera and can provide higher resolution than the rest (105). Some early approaches using stereo vision have been presented by Se et al. (114). An important work using monocular vision on real-time V-SLAM was achieved by Davison (32), where he presented handling of uncertainty in Bayesian framework. He shows robust localisation using sparse map of landmarks.

Algorithms of V-SLAM can be classified into four main categories: sparse SLAM, dense SLAM, feature-based and feature-less (11). Sparse SLAM uses a small selected subset of the pixels in image frame (43), (42), compared to dense SLAM which uses most of the pixels in each received frame. Dense SLAM thus require the use of GPU and more powerful hardware and the method of map generation is different. Feature-based also known as indirect SLAM attempts to extract simple features such as corners, edges and feature descriptor (SIFT, ORB, FAST, etc.) (108), (4) from images and then make use of them to build the map and locate the camera. Feature-less or Direct SLAM, in contrast make use of pixel intensities directly without extracting intermediate features. The basic modules of V-SLAM are explored in details in (129).

In chapter 6, we will discuss the state-of-the-art ORB-SLAM for monocular SLAM.

2.4 Perspective-n-Point Problem

Pose estimation is defined as the position and orientation of an object with respect to reference points in space. In computer vision, this is referred to the position of the camera relative to a 3D structure in the scene. To estimate the pose of a calibrated camera, a number of 3D to 2D correspondences from the 3D world to the image plane is required. This problem is referred to as perspective-n-point problem (52) and was originated from camera calibration (3). The problem describes a rigid body transformation by solving a polynomial system of equations. For n = 3, the problem is referred to as p3p.

A huge amount of work have been addressing this problem, that until now, there is no fast and optimal solution. Up until now only closed formed solutions were presented. Fischler and Bolles were the first to summarize the problem (46). They notice that there are at most four solutions for the p3p. Hung et al (62) presented an algorithm for computing the 3D coordinates of the center of the points relative to the image frame, while DeMenthon et al (33), presented analytical solution with approximations to obtain simpler solutions. A great background for the p3p is presented in (52), (136).

P3P is a non-iterative method in which the intrinsic parameters are assumed to be known. The solution need to be refined by a non-linear estimation. This makes its solution non accurate and thus more reference points are needed. This keeps the solution restricted to finding the external parameters describing the pose of the camera. In contrast, the pnp problem consider finding the full solution which encloses the intrinsic and extrinsic parameters. In practice, pnp is usually combined with RANSAC in order to remove outliers (100). Some solutions for the PnP are Uncalibrated PnP or UPnP (127), Direct Linear Transform and Efficient PnP or EPnP (77). F.Moreno et al.'s (77) introduced the EPnP method which

provides efficient solution assuming the camera parameters are known with n set of correspondences.

3 Image Formation and Perspective Camera

In the 5th century, the philosopher Mozi in ancient China discovered the formation of an image from a reflected light passing through a very tiny hole in the wall (30). A pinhole on the surface of a wall in the room will cast a dimmed inverted image of the outside world on the opposite wall of the room. In reality, the pinhole of a camera consists of a chamber, with a small hole in front. This process of image formation through a pinhole involves a perspective projection of the 3D world points onto a 2D image plane. From images, shapes, sizes and pose of an object in the 3D world can be deduced. In this section we discuss how images are formed and captured and transformation from 3D to 2D image plane is explored here.

3.1 Projective Geometry

Geometry in general focuses on studying shapes and sizes of objects in different dimensions and spaces. It started with Euclid postulating the foundation of Euclidean geometry in the series books of Euclid's Elements (59). Geometry has then expanded to other non-Euclidean geometries like projective geometry and affine geometry. These are the most famous nontrivial types of geometries and are complementary. Euclidean geometry involves in distances and angles. Conversely, affine geometry introduces methods of linear algebra into geometry, before projective space is constructed and studies objects geometry whose shapes are preserved relative to affine transformations (111),(56), (102). Finally, projective geometry deals with how lines and shapes are projected into space and studies objects whose shapes are preserved relative to projective transformations (111). Throughout the work we will deal with all the three forms of geometry. We will deal with Euclidean space to define distances and measurements and use affine space as fundamental link between projective and Euclidean space to describe linear transformation and mapping between images.

One of the basic reasons for discussing projective geometry is for its application to the geometry of Euclidean space dealing with distances and angles. The second reason is because it provides mathematical formulation to describe projection between different spaces. In that respect, projective geometry arises interests in vision domains, particularly in computer vision and computer graphics which involves camera modeling (56). Camera model and its associated transformations are based on mathematical structures of projective geometry (56). Hence, formation of 2D images from its corresponding in the 3D world is defined and achieved theoretically by means of rules from this type geometry.

3.2 Type of Projection

The generation of a flat figure of an object is made through projection. In general, a projection is a transformation that transforms points in mD-space to nD-space, where n is smaller or equal to m. For example a 3D projection maps points in 3D-space to a projective space or plane by projecting all the points towards the origin or source of projection called Center Of Projection (COP) (17). In 3D projection, there are two main means of projecting points. Projecting 3D points into a 2D plane can be carried out by rays in form of a curve or by rays that form straight lines. Rectilinear projection describes how straight rays are projected, while the curvilinear projection represents the projection by curved rays.

When the distance between the COP and the image plane is infinite the projection is called parallel projection. In this case we talk about two sub-types known as orthographic projection, where the direction of the rays of projection is normal to view plane, and oblique projection, where the rays do not project in a normal way to the view plane. On the other hand, when the distance is finite, the projection has the shape of conic and is called central projection; in this case, the projectors are not parallel to each other (17), (30). A good illustration of the differences is to observe the image formed by the shadow of the same object from two different light sources, the sun which is infinitely far, representing the infinite focal distance and a light bulb, which is at small distance from the object. The latter type of projection is very famous and is known by the perspective projection and is often used to describe how light rays are projected in human vision and arti-

ficial vision systems. We group the types of projection in the diagram in figure 2.1

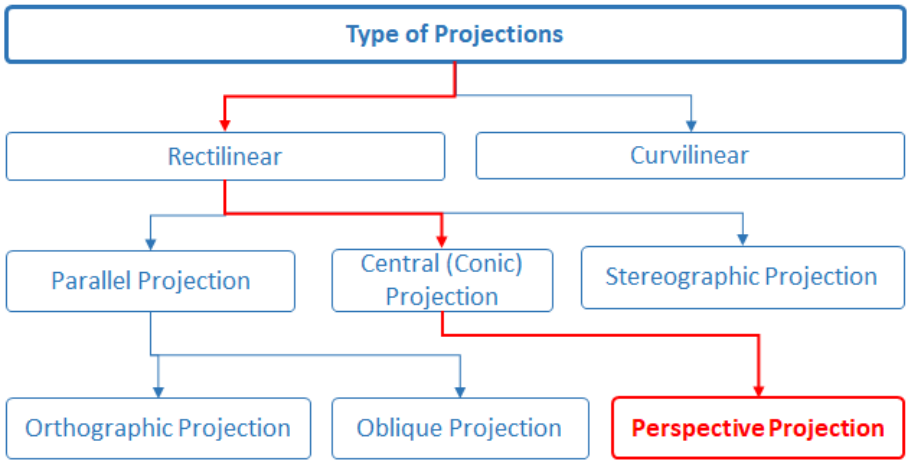


Figure 2.1: Type of projections

3.3 Geometric Property of Perspective Projection

If you look at digital computer graphics images or art work after the renaissance, one can realise distinguishing features in the development of linear perspective which respect proportions and make imagery resembles reality. Filippo Brunelleschi and Leon Battista Alberti were the first to demonstrate its principles (128) and wrote about the perspective in the exposition Della Pictura in 1435 which was based on the scientific content of classical optics in determining perspective.

In perspective view everything changes, proportions, size, shapes, the angles that we see and even overlapped objects in the scene starts to appear. Objects further away become smaller, placing an object in front of the view point can block the view of the moon while clearly the moon is much bigger than the object. Lines, even though they are parallel, seem to converge in the distance to a vanishing point, although they never actually meet.

The implication, out of perspective nature, results in the proposition of the perspective projection properties stating the following (17):

(1) Points are mapped to points

(2) Segments or lines are mapped to different segments or lines in perspective projection.

(3) Conics are mapped to different conics; circles and spheres become ellipses.

(4) Parallel lines are not necessarily parallel. In fact they meet at infinity, at a point called the vanishing point.

(5) Angles are not preserved

(6) Depth information is lost

As a consequence, the resulting equations out of the perspective projection model are often non-linear (49), (47).

3.4 Perspective Distortion

In computer vision, researchers have employed in their image analysis approximation of the perspective projection (6). The main reason for using approximation is the introduction of various distortions to the projection of the objects by the perspective projection process. John Y. Aloimonos in (6) described perspective distortion in a very nice way. According to him, these distortions are due to three main phenomenon and effects which are the distance effect, the position effect and the foreshortening effect.

Because of distance effect, objects appear smaller as they move further away from the center of projection. Object shape is also affected by the position effect; angles are not preserved but in fact, they do depend of the pose of the object. And finally the foreshortening effect creates a shape distortion that depends on the angle formed by the line of sight from the center of the object and the normal of the image plane.

We followed John Y. Aloimonos description in (6) to present these distortion throughout this thesis. A big part of this work is focused on modeling these effects in order to restore shapes of distorted objects in the scene. This part is later explored in chapter 3.

3.5 Pinhole Model

The pinhole model is a purely geometric model that describes the process of where points in the world are projected into the camera image. It is the simplest model which captures a perspective projection. Perspective projection model is mathematically convenient with the optics of camera systems (98) and is generally the ideal model for image formation (98), (49).

Assuming that the optical center is at the origin of the world coordinate system, we represent points in 3D world by $P = [u, v, w]^T$. To make the model more convenient and for simplifying the derivation of the projections, the image is considered to be between the optical center and the outside world and not behind the COP. The point of intersection between the optical axis and the image plane is called the principle point and the distance between the COP and the image plane is the focal distance, which we will denote by F throughout the work. The model in general describes the likelihood $P_r(x|w)$ of observing the projection of the point $P' = [x, y]^T$ on the image plane given that it is the projection of the point $P = [u, v, w]^T$ in the 3D world. But since our developed model is purely deterministic, we will omit this representation and focus on establishing a deterministic relationship between P in the 3D world and its projection P' on the image plane. We depict the model in figure 2.2.

We first consider the normalized model where F is equal to one. This shows how simply this model captures the perspective effect. In such a simple model, the COP of the camera is assumed to coincide with origin of the world, and the camera optical axis is assumed to be aligned with the z-axis of the world and the plane of the camera is placed in front of the center of projection to avoid inverting the image (98).

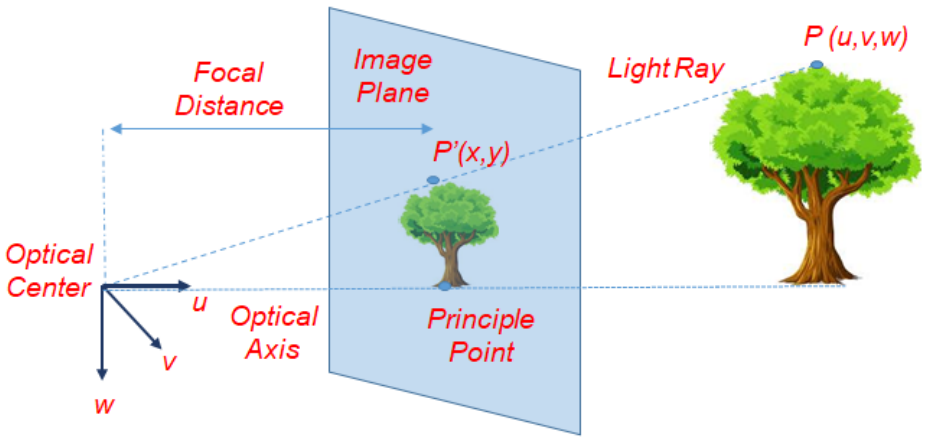


Figure 2.2: Camera Pinhole Model.

By applying the similar triangle theorem, we can write the the coordinates of the projected point P' as in equation 2.1.

$$\frac{x}{F} = \frac{u}{w}$$
$$\frac{y}{F} = \frac{v}{w}$$

and since F is equal to one,

$$\begin{aligned} x &= \frac{u}{w} \\ y &= \frac{v}{w} \end{aligned} \tag{2.1}$$

where x, y, u, v and w have the same real world unit. To make the equation linear, we change this representation from Cartesian coordinates representation to homogeneous coordinates representation, by adding a new dimension to the image points and the world points. We denote the points in homogeneous coordinates by $\tilde{P}$ and $\tilde{P}'$ as in equations 2.2.

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} F & 0 & 0 & 0 \\ 0 & F & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \tag{2.2}$$

In reality, the focal length F is set by the manufacturer, depending on the camera type and quality image.

Furthermore and to make the model realistic, the photoreceptors spacing and pixel size should be considered. These factors affect the mapping of the point P' and are represented by ϕ. The factor ϕ can also differ in the x and y-direction. This lead to equation 2.3.

$$\begin{aligned} x &= \frac{\phi_x \cdot u}{w} \\ y &= \frac{\phi_y \cdot v}{w} \end{aligned} \tag{2.3}$$

More complication can be added to the model if we consider that the image coordinate axis are not perfectly orthogonal (98). This deviation is represented by the skew parameter γ. This gives us equation 2.4.

$$\begin{aligned} x &= \frac{\phi_x \cdot u + \gamma \cdot u}{w} \cdot F \\ y &= \frac{\phi_y \cdot v}{w} \cdot F \end{aligned} \tag{2.4}$$

Finally, and in most image systems, the origin $(0, 0)$ of the image is located at the top left which requires adding a translation δ of the principal point to that position, so the final model becomes as in equation 2.5.

$$\begin{aligned} x &= \frac{\phi_x \cdot u + \gamma \cdot u}{w} \cdot F + \delta_x \\ y &= \frac{\phi_y \cdot v}{w} \cdot F + \delta_y \end{aligned} \tag{2.5}$$

The exterior parameters related to the real world are also added here since the camera is not always positioned at the center of the world coordinate system. Therefore the position and orientation of the camera are integrated in a 3x3 rotation matrix Ω and 3x1 translation vector τ. We then, express the coordinates of the world points by $P_w = \Omega \cdot P + \tau$.

Consequently, we get two sets of parameters that describe the model: The intrinsic parameters related to the internal hardware as function of $(F, \phi_x, \phi_y, \gamma, \delta_x, \delta_y)$

and the extrinsic parameters function of (Ω, τ). Putting all the intrinsic parameters in matrix form denoted by Λ we get equation 2.6, (98).

$$\Lambda = \begin{bmatrix} \phi_x \cdot F & \gamma & \delta_x \\ 0 & \phi_y \cdot F & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

Finally, we get to our complete camera transform model in equation 2.7.

$$\lambda \tilde{P}' = \Lambda[\Omega\tau]\tilde{P} \tag{2.7}$$

3.6 Distance to Target from Monocular View

In a very famous publication Saxena & Ng (112) present a probabilistic model for capturing multi-scale depth relation from a trained set of data. Other methods for estimating depth using monocular vision have been proposed (103).

Recovering depth from monocular images is a basic problem in computer vision. It requires taking into account the global structure of the image as well as the use of prior knowledge of the scene. A way to determine the distance to a preceding target of unknown size is possible using the knowledge about the height and pitch angle of the camera and the object bearing to the point where it meets the ground (75). In all the work here, we use the target geometry as prior knowledge in order to add up additional constraint to our solution. We will show here, as well a basic model of extracting the distance d from monocular view as function of the intrinsic terms considering as well that the target lies at the optical axis. This model is represented in equation 2.8

$$d = \frac{F \cdot R_{ImageResolution} \cdot S_{realscale}}{S'_{projectedscale} \cdot \phi_y} \tag{2.8}$$

Here, the focal length F is in pixel unit.

3.7 Lens Distortion

In reality, pinhole cameras rarely exist. An optical lens is usually added on top of the camera sensor. Optical lenses have one essential job which is to bring more light inside the chamber into the image sensor by focusing the light beams. Otherwise, images become too dark and blurry.

It sounds simple, but if we look at its basic anatomy in figure 2.3, there are big loads of different glasses and pieces inside and outside. Lenses are way more complex and there are a lot of sciences and researches that go behind their design and manufacturing in order to create perfectly sharp non-distorted shapes across the entire photo.

In practice, this leads to an additional distortion added to the pinhole model. Different sorts of geometric distortions, that can be induced because of lenses, are listed here (94), (50): Radial, Barrel, Decentering, Thin-Prism, Pincushion, Mustache, etc. Hence, lens distortion should be taken into consideration. One crucial distortion that needs to be addressed is explored here.

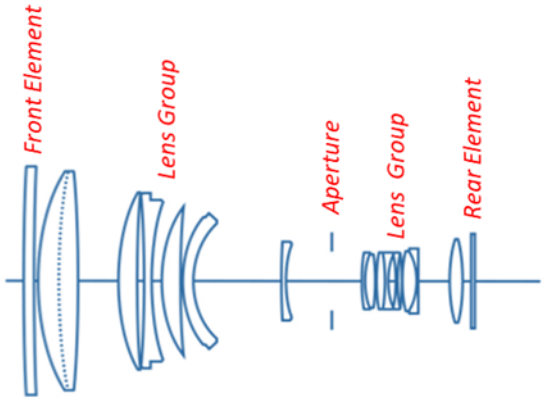


Figure 2.3: Camera Optical Lens

Radial distortion makes images become warped non-linearly which mainly affect objects in the image scene, depending on their position with respect to the center of the image. Because of radial distortion, straight lines are no longer straight in the image. Radial distortions are usually modeled in a polynomial form as

function of the distance r to the center of the image. The distortion model is shown in equation 2.9.

$$\begin{aligned} x &= x_d(1 + \beta_1 r^2 + \beta_2 r^4) \\ y &= y_d(1 + \beta_1 r^2 + \beta_2 r^4) \end{aligned} \tag{2.9}$$

where (x_d, y_d) represent the distorted projections and (x, y) are the correct coordinates of the projection, and β_1 and β_2 are the parameters that control the degree of distortion.

3.8 Camera Calibration

Camera calibration is the process of learning the intrinsic parameters of the camera. Before doing any experimental work, we had to extract our intrinsic parameters in order to feed our developed model with these fundamental parameters. Furthermore, this process is essential for eliminating lens distortion which could greatly affect our experimental results.

Theoretically, this process aims at estimating the maximum likelihood P_r of $(x_i|w_i, \Lambda, \Omega, \tau)$, given a known object in space, with I distinct 3D $\{w_i\}_{i=1}^{I}$ coordinates points knowing their 2D corresponding projections $\{x_i\}_{i=1}^{I}$ in the image. This process is depicted in equation 2.10

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} \left[\max_{\Omega,\tau} \left[\sum_{I}^{i=1} log[Pr(x_i|w_i, \Lambda, \Omega, \tau)] \right] \right] \tag{2.10}$$

Practically, we use checker board (20x12) with known square sizes for our 3D reference points in the world.

3.9 2D and 3D Transformation

Transformation is a manipulation process on points in space for the purpose of mapping or operating on these points.

3.9.1 Rigid Transformation

In this section the basics of Rigid transformation is covered. These transformations are defined such that distance, angles and shapes of the object are preserved. In the context of the work, we use these transformations to move the camera in 3D space by applying rotation, translation or reflection.

Translation operation creates a linear displacement of the object in 3D space. Let the point be P to represent the center of an object, by the vector $[x, y, z]^T$. The translation of the point P to P' by t can be represented as in equation 2.11 and can be defined in homogeneous coordinates by the matrix T as shown in equation 2.12.

$$P' = P + t = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2.11}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

Rotation transformation performs rotation operation on the rigid body in Euclidean space. Rotation in 3D are defined in homogeneous coordinates depending

on the axis of rotation, as shown by the transformation matrix in 2.13.

$$
\begin{aligned}
R_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
R_y(\beta) &= \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
R_z(\gamma) &= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2.13}
$$

Any rotation can be obtained by using matrix multiplication following fixed angles convention $R_{XYZ} = R_Z(\alpha) \cdot R_Y(\beta) \cdot R_X(\gamma)$, or for example Euler angles or by using Euler parameters (Quaternions).

3.9.2 Non-Rigid Transformation

Scaling perform operation on the size of the object. Object loses it rigidity and its size is no longer preserved. Vector P can be scaled by S_x, S_y, S_z in the axes x, y and z direction using the the matrix S in equation 2.14

$$
S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{2.14}
$$

We can group all the above transformations in one matrix form T as follow 2.15:

$$T = \begin{bmatrix} RS & t \\ 0 & 1 \end{bmatrix} \tag{2.15}$$

3.9.3 Image Transform

We show in figure 2.4 the types of image transformations. We will use most of these transformations for the purpose of applying perspective correction, to define our projective rotation concept and to apply image rectifications by recovering our transformation. We list below all of the transformation models in homogeneous coordinate system as a reference.

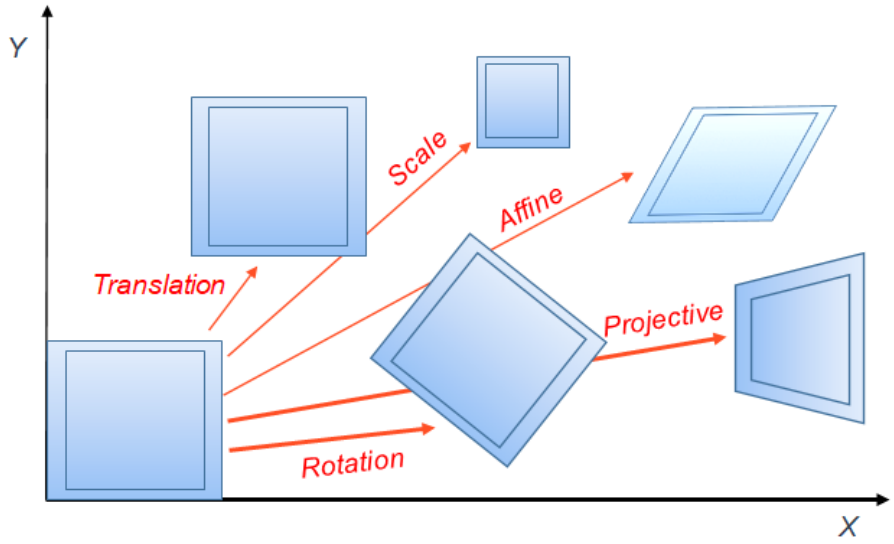


Figure 2.4: 2D Image transformations.

$$T_{(translation)} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad R_{(Rotation)} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$S_{(Scale)} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad A_{(Affine)} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$
$$P_{(Projective)} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

An important thing here is that all these transformations can be combined by matrix multiplication. From the number of parameters in every transformation, we deduce the degree of freedom for every system and thus the number of parameters to solve our transformations.

3.9.4 Homography Transformation

A homography is a transformation between two images taken from the same center of projection. It is defined by 8 parameters as shown in equation 2.17.

$$H_{(Homography)} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \tag{2.17}$$

Chapter 3

Perspective Projection and Distortion Correction

Contents

1 Chapter Summary

Figure 3.1: Perspective Distortion

There are two types of distortions that can affect object shapes and sizes in the scene, when projected into an image plane: optical and perspective distortion (130), (133). The former is related to the design of the optical lenses, which plays an important role in bring more light into the camera sensor. Lenses are way more complex and are discussed in more details in chapter 2, section 3.7.

This full complexity of the lenses is beyond the scope of the work in this chapter and thesis. Our main focus in this chapter is on the latter type of distortion, which is perspective distortion. For this reason, mainly in most experimentation and work that follow, we will consider our system to be composed of a simple pinhole model with no additional lens.

Perspective distortion refers to a state of spatial perception, which has an important role in photography. Understanding this distortion has long been an interesting topic for artists and designers, and has also been a subject of research for image diagnostic, monitoring and camera pose estimation.

The work in this chapter covers mainly perspective distortion with specific focus on foreshortening effect and provides deep analysis from the geometric and analytic point of view, to come up with distinct correction methods. Distinct cases are presented in separate sections for different geometric shapes:

(1) We first start by exploring the effect on line segments, from which we break-out the end resulting equations into different factors related to its proper distortion, to come up with proofs and correction methods for every factor. We end this section by analysis and discussion of the results.

(2) We then expand the study for circular and spherical targets where we introduce eccentricity correction method followed by simulation. We finish our discussion with 3D relative position estimation approach.

In the next section, we give a brief definition on perspective distortion of objects in the scene after projection.

2 Perspective and Foreshortening distortion

2.1 Perspective Effects

Figure 3.2: Perspective Effect on the left and right fist.

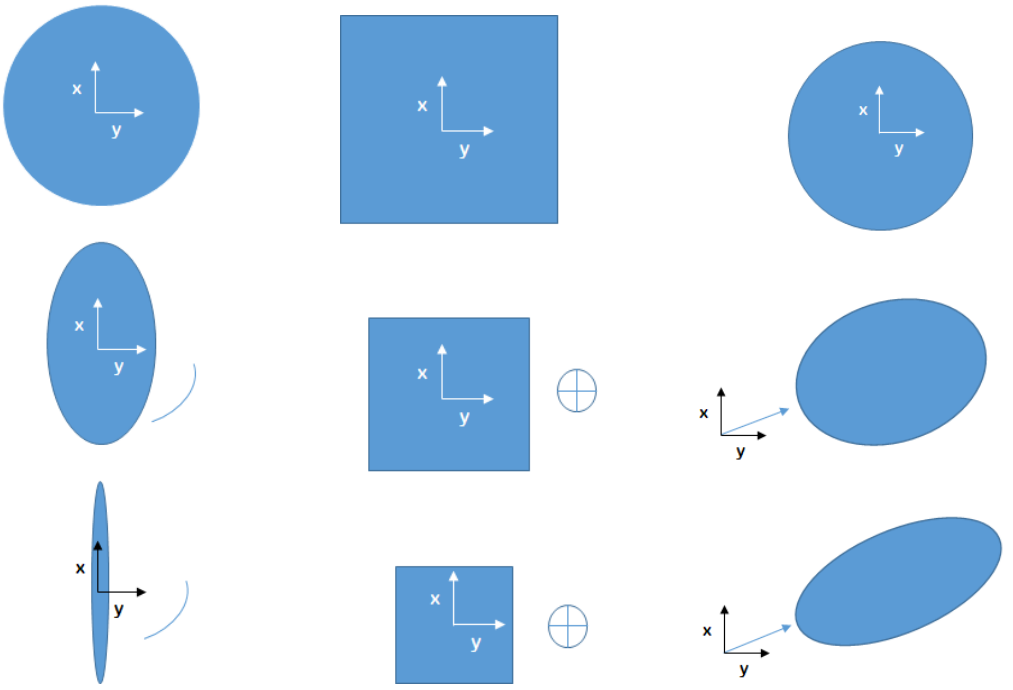
Perspective distortion is a function of distance between the camera and the subject in the scene, which causes the target subject to appear unnatural. It is

strictly dependent of the position of the camera with respect to the target object within the scene, as in figure 3.2. This phenomenon has to be necessarily taken into consideration during monitoring of industrial manufacturing processes and it is relevant in most of the cases to understand and eliminate this kind of distortion.

This aberrant appearance of the targets in the scene is induced by three main phenomenons of perspective effect. These three main phenomenons are the distance effect, the pose effect and the foreshortening effect. All three effects are illustrated in figure 3.4. Because of distance effect, objects appear smaller as they move further away from the center of projection. Object shapes are also affected by the position effect; angles are not preserved but in fact, they do depend from the pose of the object. And finally, the foreshortening effect create shapes distortions that depend on the angles formed by the line of sight from the center of the objects and the normal of the image plane.

Figure 3.3: Perspective Effect on parallel lines

The implications out of these distortions result in the propositions of perspective projection stated in chapter 2, section 3.3.



(a) Pose Effect (b) Distance Effect (c) Foreshortening Effect

Figure 3.4: Forms of Perspective Effects seen from the image plane

Figure 3.3 shows the implications of the perspective distortion properties on straight and parallel lines, on conics, on angles and depth.

As a consequence, the resulting equations, out of the perspective projection models, become complicated and often are non-linear (49), (47) as we will see in the later sections.

In the following section, we will explore the foreshortening effect which is one relevant effect of perspective distortion.

2.2 Foreshortening Effect

As we discussed earlier in chapter 2 there are three forms of distortions resulting from perspective projection (6); one of them is called foreshortening distortion.

Foreshortening effect is a major distortion that has direct impact on how the shapes and sizes of the objects appear when projected into a perspective plane. This effect creates shape distortion that is related to the angle formed by the line of sight from the center of the object and the normal to the image plane.

In many cases, the knowledge of the shape and size of the projection prior to distortion is a necessity, for example, for estimating the pose or distance of that object with respect to the perspective view point. The process of restoring or estimating the original projected form without distortion, is referred to here as the foreshortening correction. This is a common problem known by the term Identification of Eccentricity Error, where eccentricity reflects how the projected center of a target object is deviated due to foreshortening effect and others.

3 Distortion Effects on Line-Segments

3.1 Projection of Centered Line-Segments

For better understanding of the problem, we begin an investigation and explore first how a line-segment, as a target object of a length l in space, projects onto the projection plane. And for simplicity and good comprehension in this case, the segment is considered to be in a $2D$ space (XY) and thus, the perspective plane is foreshortened to a line located at distance F from the perspective view point. This methodology can be simply extended from $2D$ to $3D$ by applying the same process on different plane, which is in this case the (XZ) plane.

Before starting to sink down deeply in all sorts of terminologies and equations, we would like to start constructing a workflow by placing our first build block into the structure. And through out the following section, we will keep adding and updating the blocks within the chart. The workflow is intended to support the reader in pursuing our methodology and thinking. Figure 3.5 illustrate the first block.

We aim here to extract the targets center's position and orientation, free from all

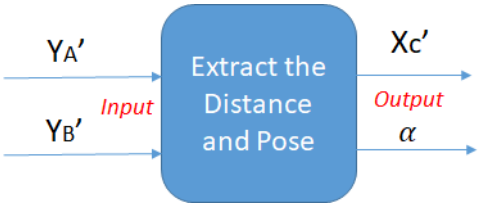


Figure 3.5: Workflow

distortions, having our image projected points of the subject.

Considering the line-segment represented by two end points $A = [x_A, y_A]^T$ and $B = [x_B, y_B]^T$ as shown by the figure 3.6, and is of a known length l.

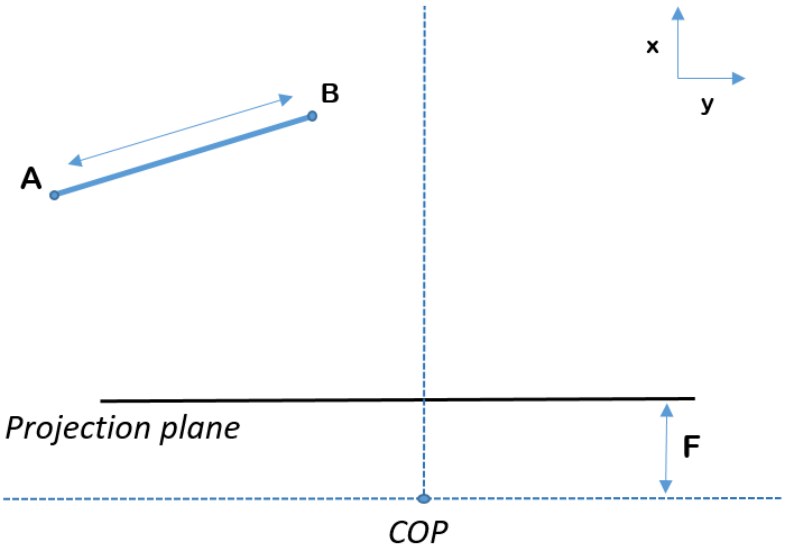


Figure 3.6: Line Segment in space

If the segment is placed to be coincident at its mid-point C with the perpendicular to the image plane and is tilted by an angle α from this plane, as shown by figure 3.7, the oblique projections of A and B towards the Center Of Projection can be calculated from similarity and perspective projection postulate following eq. 3.1 and eq. 3.2. We represent the projection of A and B by $A' = [x_{A'}, y_{A'}]^T$ and $B' = [x_{B'}, y_{B'}]^T$ simultaneously and by α the tilted angle of the segment from the image plane.

Since x_C represents distance of the center of the segment from the COP, x_C can

be calculated from 3.3.

Proof. Let $x_C = (x_A + x_B)/2$ and $AC = L/2$, a substitution of eq. 3.1 and eq. 3.2 in x_C will result in eq. 3.3. x_C here represents the distance to the center of the segment $[AB]$ from the COP. □

$$y_{A'} = \frac{F}{x_A} \cdot AC \cdot \cos(\alpha) \tag{3.1}$$

$$y_{B'} = -\frac{F}{x_B} \cdot BC \cdot \cos(\alpha) \tag{3.2}$$

$$x_C = \frac{F}{2} \cdot \left[\frac{y_{B'} - y_{A'}}{y_{B'} \cdot y_{A'}} \right] \cdot \frac{L}{2} \cdot \cos(\alpha) \tag{3.3}$$

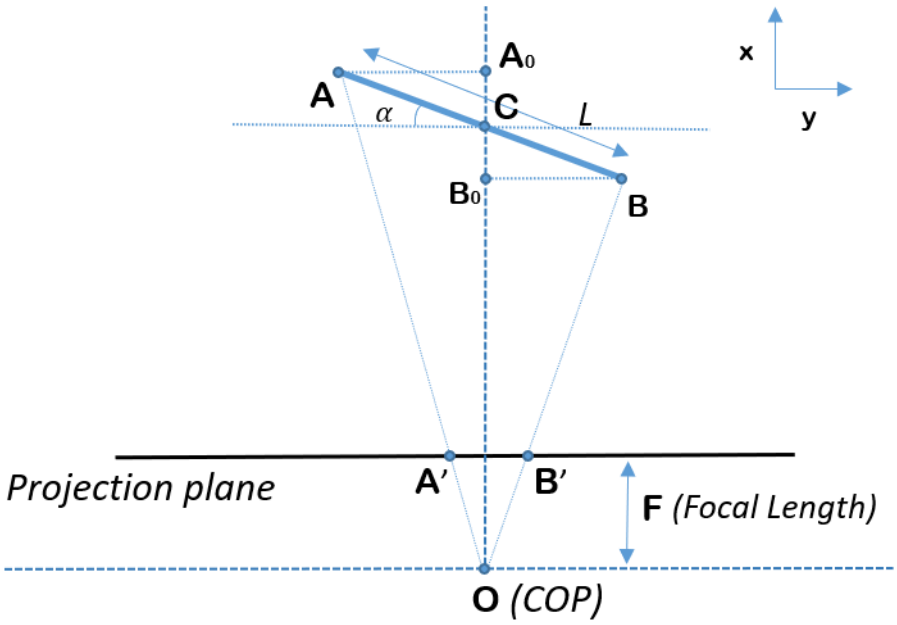


Figure 3.7: Centered Line-Segment

Furthermore, and for this particular case, only where the center of the target is known and lies at the normal to the image which intersects with COP, the tilted angle α can be derived as depicted in eq. 3.4.

$$\alpha = \arctan\left(\frac{F}{2}.\frac{y_{B'} + y_{A'}}{y_{A'}.y_{B'}}\right) \qquad \alpha \in [-\tfrac{\pi}{2} \quad \tfrac{\pi}{2}] \tag{3.4}$$

Proof. From perspective projection $x_A = \frac{F}{y_{A'}} \cdot AC \cdot \cos\alpha$ and $x_B = -\frac{F}{y_{B'}} \cdot AC \cdot \cos\alpha$ in addition,

$$x_A - x_B = AB \cdot \sin\alpha$$

moreover

$$\begin{aligned} x_A - x_B &= \left(\frac{F}{y_{A'}} + \frac{F}{y_{B'}}\right).AC.\cos\alpha \\ &= \left(F.\frac{y_{B'} + y_{A'}}{y_{A'}.y_{B'}}\right).AC.\cos\alpha \end{aligned}$$

which implies

$$AB \cdot \sin\alpha = \left(F \cdot \frac{y_{B'} + y_{A'}}{y_{A'} \cdot y_{B'}}\right) \cdot AC \cdot \cos\alpha$$

implies

$$\tan\alpha = \frac{F}{2} \cdot \left(\frac{y_{B'} + y_{A'}}{y_{A'} \cdot y_{B'}}\right)$$

implies

$$\alpha = \arctan\left(\frac{F}{2} \cdot \frac{y_{B'} + y_{A'}}{y_{A'} \cdot y_{B'}}\right)$$

□

We can thus write any point P_{seg} on the segment in the camera or COP frame as $P_{COP} = \Omega \cdot P_{seg} + \tau$. Whereas in matrix representation form, we can write the full pose of the centered segment in the camera or COP coordinate frame as in eq. 3.5, where, Ω is the rotation matrix and τ is the translation vector.

$$\xi_{(x_c,y_c,\alpha)} = \begin{bmatrix} \Omega & \tau \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & x_c \\ \sin(\alpha) & \cos(\alpha) & y_c \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

The position of the center C of the segment in the COP coordinate frame becomes as in eq. 3.6.

$$C_{COP} = \begin{bmatrix} \Omega & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & x_c \\ \sin(\alpha) & \cos(\alpha) & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.6}$$

In this section, we derived the y-coordinates of the projected segment along with its distance to the center, considering that the segment is centered. We then generated the form of the full pose of this segment in camera frame. What we will do next is modeling our distortion factors and we will show that these projections hold for only two distortion factors excluded from the foreshortening distortion. This is because the line lies at the center.

3.2 Pose and Distance Factor and Correction Methods

From the previous section, eq. 3.1 and eq. 3.2 show that the projection of any centered segment is not only dependent of the segment distance from the COP, but also from the tilted angle α.

We define our distortion factor to be the ratio of the projected length over the real length of the segment. We denote this factor by the small mu and is belonging to $[0, 1]$. Since the segment $[AB]$ is centered with respect to the COP we can write eq. 3.7.

$$\mu = \frac{|\, y_{B'} - y_{A'}|}{L} \qquad \mu \in [0 \;\; 1] \tag{3.7}$$

this yields to eq. 3.8

$$\mu = \underbrace{\left|\frac{4 \cdot x_C \cdot y_{B'} \cdot y_{A'}}{F \cdot L^2}\right.}_{\substack{\text{Distance} \\ \text{Distortion Factor}}} \cdot \underbrace{\left.\frac{1}{\cos(\alpha)}\right|}_{\substack{\text{Position} \\ \text{Distortion Factor}}} \tag{3.8}$$

The right hand side of eq. 3.7 can be split into two factors where we denote the right factor, which we call Distance Distortion factor (DDF) as μ_1 and left factor, which we name Pose Distortion Factor (PDF) as μ_2. It is now clear that when $\alpha = 0$, meaning that the line lies in parallel to the image plane, the PDF is equal to one. In contrast, when X_C is equal to F, meaning that the target is located at the image plane, the DDF is equal to one. This yields to say that our μ factor is equal to one when no distortion is implied and can be written as function of both distortions:

$$\mu = \mu_1 \cdot \mu_2 \tag{3.9}$$

Since the segment $[AB]$ is centered, foreshortening distortion has no effect on its projection $[A'B']$ as in analogy to eq. 3.9. The only existing distortions in this case are the pose and distance effects.

As a conclusion, the projected length of segment in μ, which is derived from 3.3, holds information on both distortions throughout the two factors that are named here, pose and distance factors, as it is depicted. In fact, as the center of the segment moves away from the normal to the perspective plane, the foreshortening distortion starts appearing and provoking the projected length $[A'B']$ of the segment $[AB]$ to change non-linearly. In this case, we can notice that the derivation of the distance from the center of the segment C to the COP, using the previous equations 3.3, has to hold for other factors due to the added foreshortening distortion to $[A'B']$, which we will see in the next section.

Furthermore, cognising the tilted α allows to compensate for the pose effect, while knowing the real length l of the segment, will add up more constraint on the distance equation 3.3. This in turns, allows eliminating the distortion coming from

distance effect simultaneously, when calculating the 3D pose of our target.

In reality and in most cases, the reversed problem matters most; the interest is to locate the segment in space where only the projected distances onto the perspective plane, or image plane, are known. Clearing out all the distortions makes it possible, for example, to calculate accurately the position and orientation of the target object with respect to the COP. In the next section we will explore the derivation of foreshortening factor when the segment lies anywhere in space. We follow the same approach and extend the study later to different target shapes for the purpose of localising the object in space, knowing only the projected geometry onto the perspective plane.

3.3 Projection of Non-Centered Line-Segment

Our goal here is to derive the target's projection excluded from the foreshortening error as if the object lies at the center of the field of view, or at the center of the perpendicular to the perspective plane, from the center point. In this process, and since we are still looking at the problem from the homogeneous 2D perspective only, a rotation of the segment $[AB]$ is performed towards the COP around the perpendicular axis where $[AB]$ lies. These rotations are based on the position of the projection which are our only available inputs. The below flowchart 3.8 depicts our intention.

Considering the new segment to be represented by the points D and E, and to be of center G. Since the projection of the center of the target in most cases is not known, a distinction between different rotations process is considered in order to select the proper rotation starting point. These processes are depicted below:

(1) Rotation from the target's center. This requires that the mid-point of the target to be known.

(2) Rotation from the mid-point of the projected target.

(3) Rotation from the mid-arc of the projection.

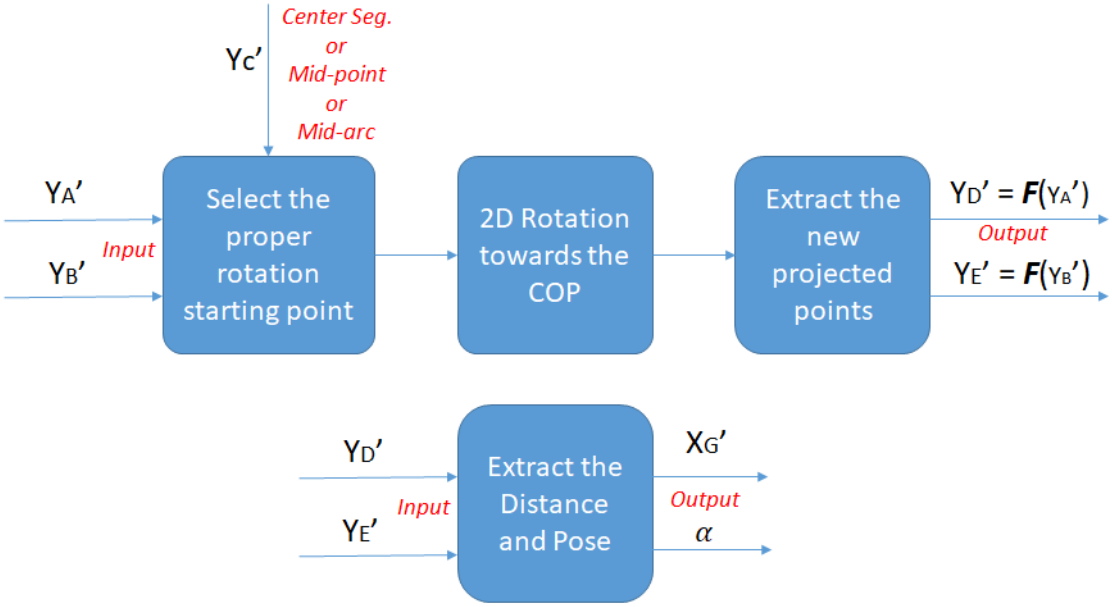


Figure 3.8: Workflow

Within the context of this thesis, these kinds of rotation will be called projective rotation throughout the work as we are rotating around the COP using only projected points as input. Although we perform a rotation around the COP, projective rotation provokes the rotated projection to translate in the same plane and not to rotate in the 2D image plane instead.

In all of the above three cases listed, the rotation is considered to happen around the $Z - axis$ when working in the (XY) plane, and around the $Y - axis$ when working in the (XZ) plane, as shown in the figure 3.9. Furthermore, cases (2) and (3) can be seen as complementary cases since they solve the same issue but one can have preferences in some occasions, as we will explore later.

Since there are cases where our target center is not known, we begin in the next section by locating all possible centers locations.

3.3.1 Target Center Location

Before proceeding with rotation cases, and considering that the center-point of the target located anywhere in space is not known, it is important to picture

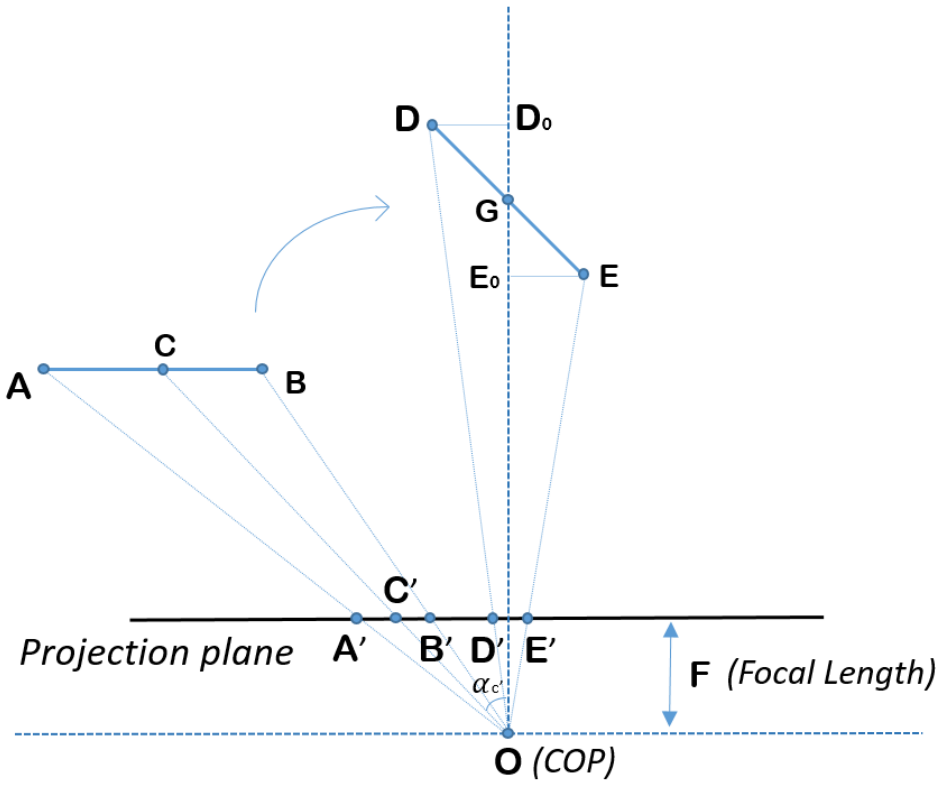


Figure 3.9: Rotation towards the COP

where that center could be, in order to perform a proper rotation that could eliminate the distortions.

Furthermore, deriving the angle of orientation of the segment from the projective plane becomes uncertain. The reason is, that the tilted α of that segment does depend from the location of the center point and not from its projection as we will show.

What we will show later as well is, that for unknown centers, the foreshortening effect cannot be restored for a line-segment except for two special cases. But fortunately as well, it turned out that these cases are very useful when dealing with circular and spherical targets as it will be shown in the circular and spherical target section.

We start next by modeling the location of the true center. A step by step demon-

stration is presented depicting the possible position of the centers for a particular projection.

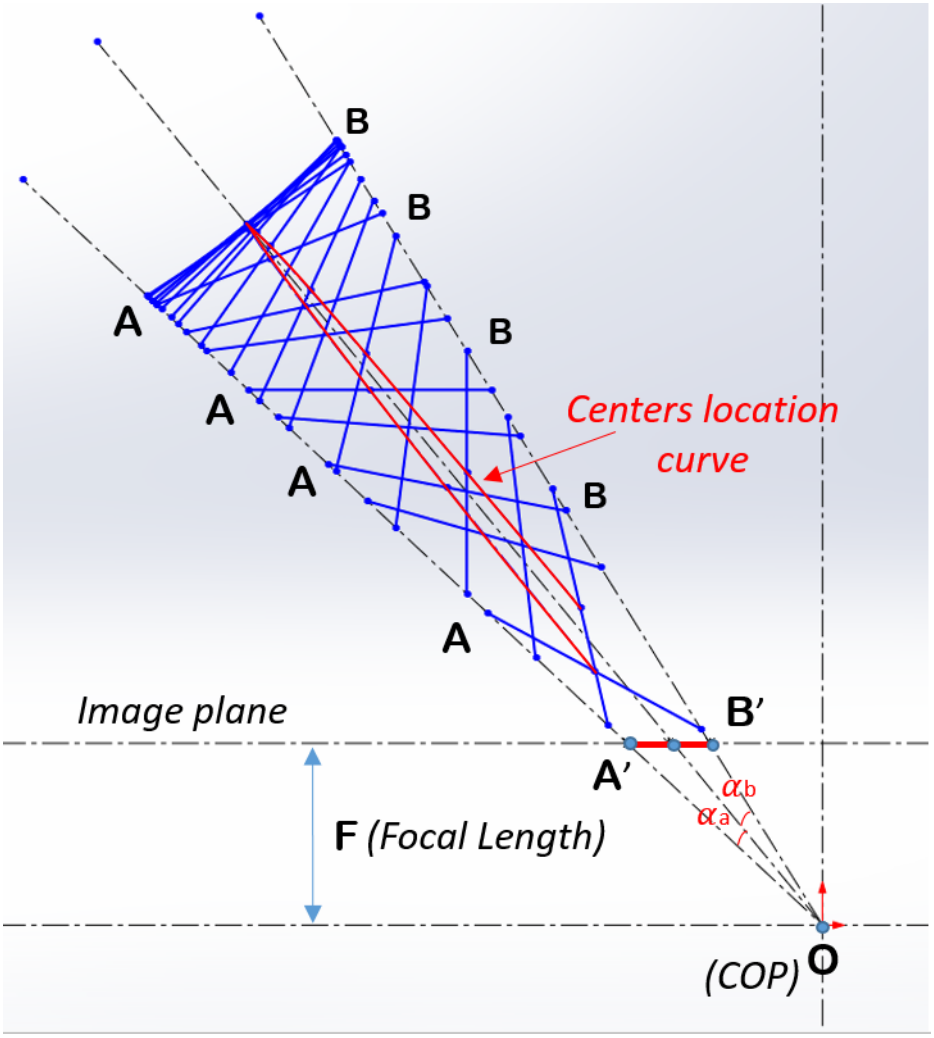


Figure 3.10: The Centers location curve of the same segment of length having the same projection but different tilted angle from image plane

Considering a line segment $[AB]$ of a length L. Figure 3.10 shows that for any particular projection $[A'B']$, different position of the same segment $[AB]$ can be originated. All the segments in this figure are of the same length l but what can be noticed is that every possible unique position has its different tilted angle α. α is the angle formed by the segment and its projection which lies on the image plane. This unique position, of every segment, does depend of the angle α. Furthermore, a plot of the position of all the centers of the segments is rendered as the centers location curve, and is depicted in this figure. The center location

curve has the shape of a parabola and has the line of sight as the axes of symmetry. Finally, what we noticed are some very notable features. There are two remarkable and important features that can be extracted from this pattern.

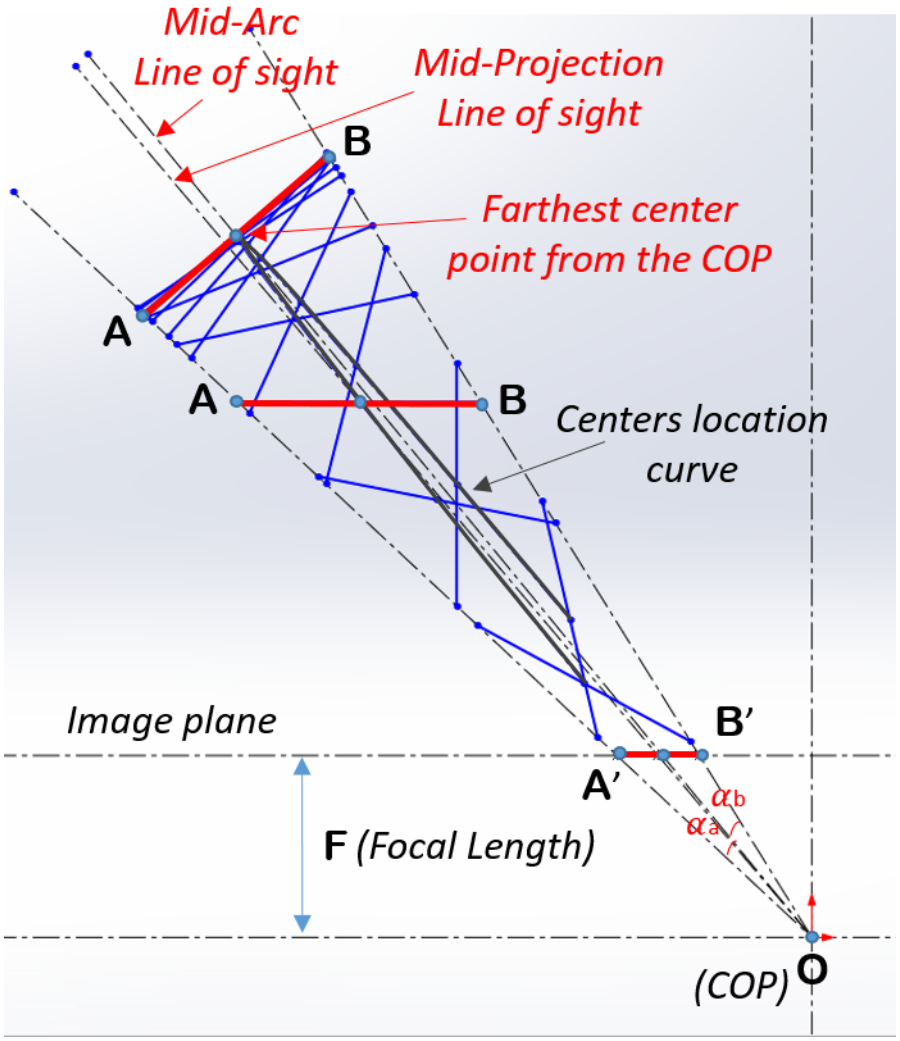


Figure 3.11: The Centers location curve of the same segment of length having the same projection but different tilted angle from image plane

Figure 3.11 depicts those features. In this context, these remarkable features are referred to as aspect (1) and aspect (2).

Aspect (1): The first distinctive feature is at the maximum of the center location curve. This center corresponds to the farthest possible segment from the center of projection. This segment represented by $[AB]$ is perpendicular to the mid-arc line of projection and is the only segment at which its mid-point is

coincident with the mid-arc line, as it is shown by the centers location curve in the figure 3.11. This implies that the projection of its center is also coincident with mid-arc line.

Aspect (2): Another remarkable aspect is in the only segment that is parallel to the image plane, with α equal to zero. This segment $[AB]$ has its mid-point coincident with the axis that intersects with the mid-point of the projected segment $[A'B']$ and the COP.

These are two important observations. In a later section and chapter, these observations will be used as building blocks for establishing the concept of eccentricity elimination and our marker-based localisation approach.

3.3.2 Projective Rotation from the Target Center

As we talked earlier, we distinguish between three rotation cases. We start here by considering the first case where we know the projected coordinate of the mid-point of the segment.

Let A', B' represent the projection of the target segment bounded by A and B, and C' the projection of its mid-point C on the image plane. What we are searching for is a relationship between these projections and the projections of that same segment after being rotated towards the center. Meaning that, after rotation, the center C of the real segment should be coincident with focal axis, and its projection is coincident with the center of image. This makes the projected segment to become free from foreshortening effect.

Since after rotation the angles between the three projected points do not change, the new projections after rotation can be derived in a straight forward way after calculating the angles between the vectors $\vec{OA'}$, $\vec{OC'}$ and $\vec{OB'}$. If we let $\alpha_{A'C'}$ to be the angle $\angle(\vec{OA'}, \vec{OC'})$ and $\alpha_{B'C'}$ to represent the angle $\angle(\vec{OB'}, \vec{OC'})$, since these angles remain the same after rotation, the new projections y'_D and y'_E are derived in eq. 3.10 and eq. 3.11 where E', G' and D' are the new projected points

respectively, and after rotation.

$$y_{D'} = F \cdot \tan(\alpha_{A'C'}) \tag{3.10}$$

$$y_{E'} = F \cdot \tan(\alpha_{B'C'}) \tag{3.11}$$

We note here that y'_D and y'_E can also be written as in eq. 3.12 and eq. 3.13

$$y_{D'} = F^2 \cdot \frac{y_{A'} - y_{C'}}{F^2 + y_{A'} \cdot y_{C'}} \tag{3.12}$$

$$y_{E'} = F^2 \cdot \frac{y_{B'} - y_{C'}}{F^2 + y_{B'} \cdot y_{C'}} \tag{3.13}$$

Proof. Let $\alpha_{A'C'}$ be the angle $\angle(\vec{OA'}, \vec{OC'})$ then $\alpha_{A'C'} = \alpha_{D'} = \alpha_{A'} - \alpha_{C'}$ (after rotation). Or $\alpha_{c'} = (\alpha_{A'} + \alpha_{B'})/2$ where $\alpha_{A'} = \arctan(y_{A'}/F)$ and $\alpha_{B'} = \arctan(y_{B'}/F)$. This gives the following:

$$\begin{aligned}
\alpha_{D'} &= \alpha_{A'} - \alpha_{C'} \\
&= \arctan(\frac{y_{A'}}{F}) - \arctan(\frac{y_{C'}}{F}) \\
&= \arctan\left(\frac{y_{A'} - y_{C'}}{F + y_{A'} \cdot y_{C'}}\right)
\end{aligned}$$

which leads to

$$y_{D'} = F^2 \cdot \frac{y_{A'} - y_{C'}}{F^2 + y_{A'} \cdot y_{C'}}$$

□

In reality, our line-segment lives in 3D space. The description of projective rotation in 2D space is nevertheless important for understanding of the problem. But if we go one dimension up in space, projective rotation of points in one plane has shown to have direct impact on the projections in the added third dimension. For this reason, we would like shortly to depict this behavior and relationship by deriving $z_{D'}$ and $z_{E'}$ after projective rotation of D' and E' in (XY) plane.

Although while demonstrating equation 3.12 and 3.13 we performed a 2D rota-

tion in the (XY) plane, there was unusual change in the Z-coordinates in 3D of the points after rotation. This effect is hard to be visualized in 2D figures. Figure 3.12 (a) depicts the rotation in the (XY) plane, while in figure 3.12 (b) we show its impact from the (ZX) perspective. We point out at the points projections of D and E which represent the new position of A and B respectively, after rotation. We see here that when we rotate in 2D, the corresponding Z-coordinate of the projection, of the rotated point, decreases as we rotate towards the center. This is because the Z-coordinate of the point D, depicting the new position of the A after rotation around the Z-axis, does not change. Only the X and Y-coordinates are changing.

To calculate the new Z-coordinate of the new projected point, we use an appropriate intermediate point named D_1, as shown in figure 3.13. D_1 is the rotation of A' around the Z-axis. Since we are dealing with perspective projection, the Z-coordinate of the projection of point D_1 is the same as the Z-coordinate of the projection of D as in figure 3.13. Furthermore, we already calculated the Y-coordinate of D' from eq. 3.12. The 2D coordinates of D_1 are written below as in eq.3.14.

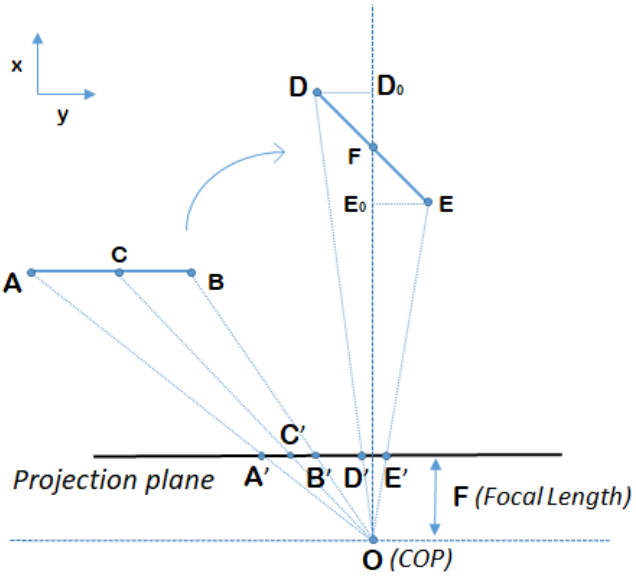
$$D_1 = \begin{bmatrix} x_{D_1} \\ y_{D_1} \\ z_{D_1} \end{bmatrix} = \begin{bmatrix} A'O \cdot cos(\alpha) \\ A'O \cdot sin(\alpha) \\ z_{A'} \end{bmatrix} \tag{3.14}$$

The vector representation and derivation of D' is shown in 3.15,

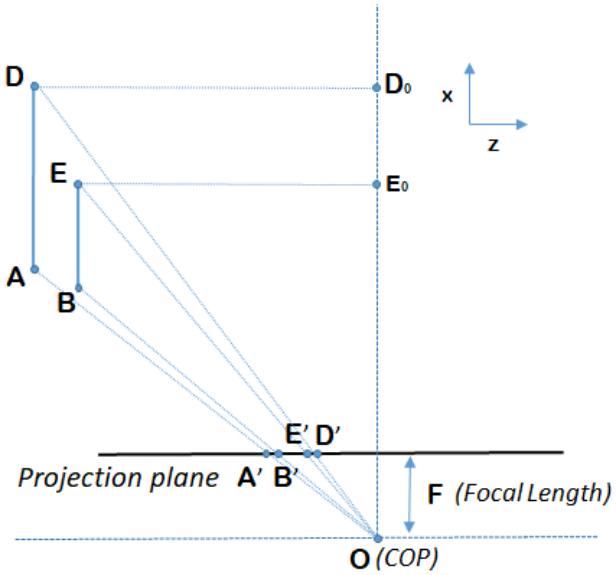
$$D' = \begin{bmatrix} \frac{x_{D_1}}{x_{D_1}} \cdot F \\ \frac{y_{D_1}}{x_{D_1}} \cdot F \\ \frac{z_{D_1}}{x_{D_1}} \cdot F \end{bmatrix} = \begin{bmatrix} F \\ y_{D'} \\ \frac{z_{D_1}}{x_{D_1}} \cdot F \end{bmatrix} = \begin{bmatrix} F \\ F^2 \cdot \frac{y_{A'} - y_{C'}}{F^2 + y_{A'} \cdot y_{C'}} \\ \frac{z_{A'}}{A'O \cdot cos(\alpha)} \cdot F \end{bmatrix} \tag{3.15}$$

and the missing Z-coordinate of D' can be obtained from eq. 3.16 where we showed its derivation after rotation in eq. 3.15.

$$z_{D'} = \frac{z_{A'}}{A'O \cdot cos(\alpha)} \cdot F \tag{3.16}$$



(a) Rotation in the (XY) plane.



(b) Rotation impact on the Z-coordinates of the points

Figure 3.12: Projective Projection impact on 3D correspondences.

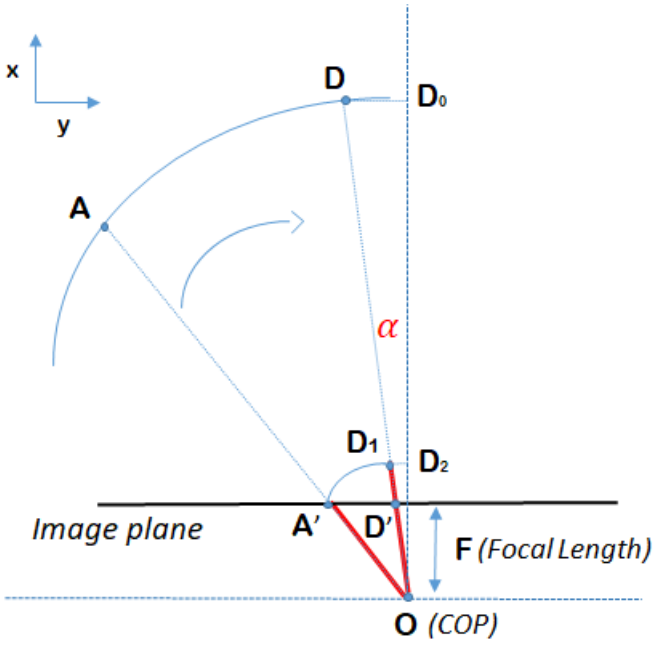


Figure 3.13: Derivation of the of the Z-coordinate from the (XY) Plane.

By deriving Eq. 3.12, 3.13 and 3.16 we have established a relationship between the projections of the points in space before and after rotation. These are fundamental equations to perform any projective rotation. We will use these equations more frequently from now on to perform any projective rotation toward the focal axis, with the help of an intermediate point which is in case C' regarding if its a center or not.

From these equations as well, we can derive the distance to the segment center prior and after rotation by substituting eq. 3.10 and 3.11 in eq. 3.1 and 3.2 respectively. Knowing that $x_{center} = (x_D + x_E)/2$, x_{center} can be treated as the distance to the center of the segment in case the rotation is with respect to the center. Its derivation is depicted in eq. 3.17.

$$x_{center} = \frac{L \cdot F^2}{4} \cdot \left[\frac{y_{D'} - y_{E'}}{y_{D'} \cdot y_{E'}} \right] \cdot \cos(\alpha) \tag{3.17}$$

The tilted angle α can also be calculated from eq. 3.4 and by adding to it the

angle of rotation α_c which yields to eq. 3.18.

$$\alpha = \arctan\left(\frac{F}{2}.\frac{y_{B'} + y_{A'}}{y_{A'}.y_{B'}}\right) + \alpha_c \tag{3.18}$$

3.3.3 Projective Rotation from the Mid-Arc of the Projection

Previously, we described the pose and distance in accordance with case (1). In this section the following case (2) and (3) are expanded from what we demonstrated before. We consider now that the center of the target is indistinct, this makes the choice of the rotation starting point to become critical.

When mid-arc rotation towards the center is performed as shown in figure 3.14, the function of the center location curve becomes centered and can be expressed by the two parametric equations 3.19 and 3.20, with the angle as the intermediate parameter. We plot these parametric equations, as shown in figure 3.15.

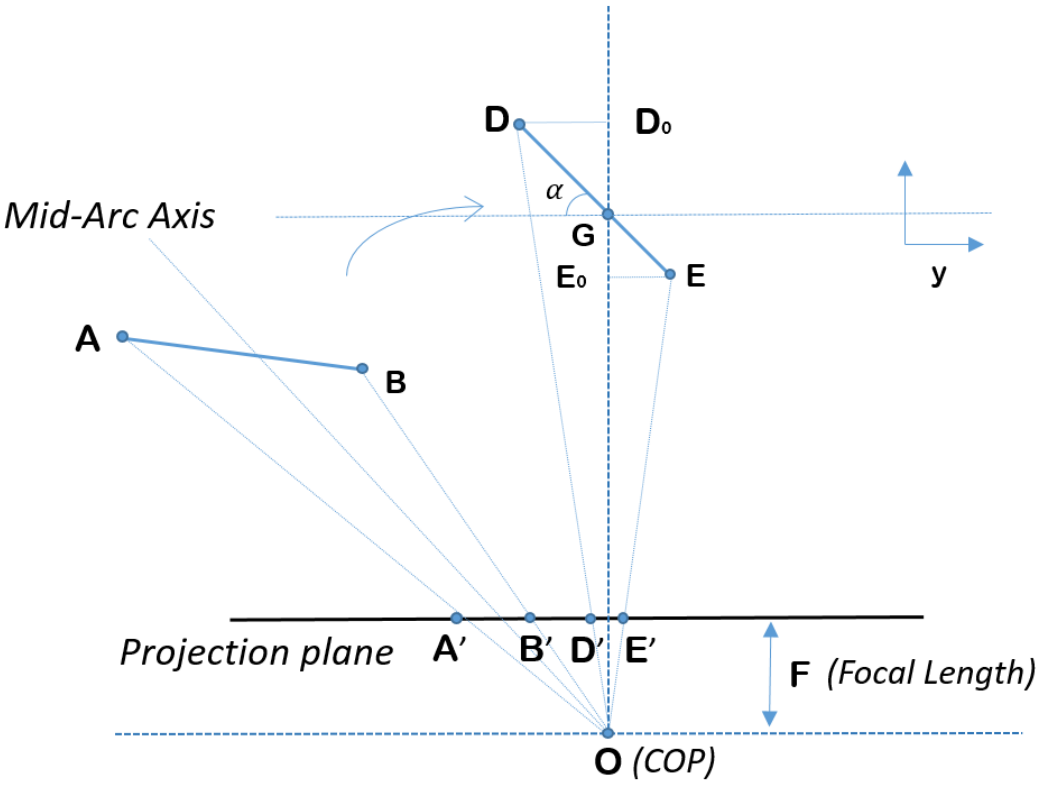


Figure 3.14: Rotation from the Mid-arc towards the center

$$x_c = \frac{L \cdot F}{2 \cdot y_{A'}} \cdot \cos(\alpha) \tag{3.19}$$

$$y_c = \frac{L \cdot y_{A'}}{2 \cdot F} \cdot \sin(\alpha) \tag{3.20}$$

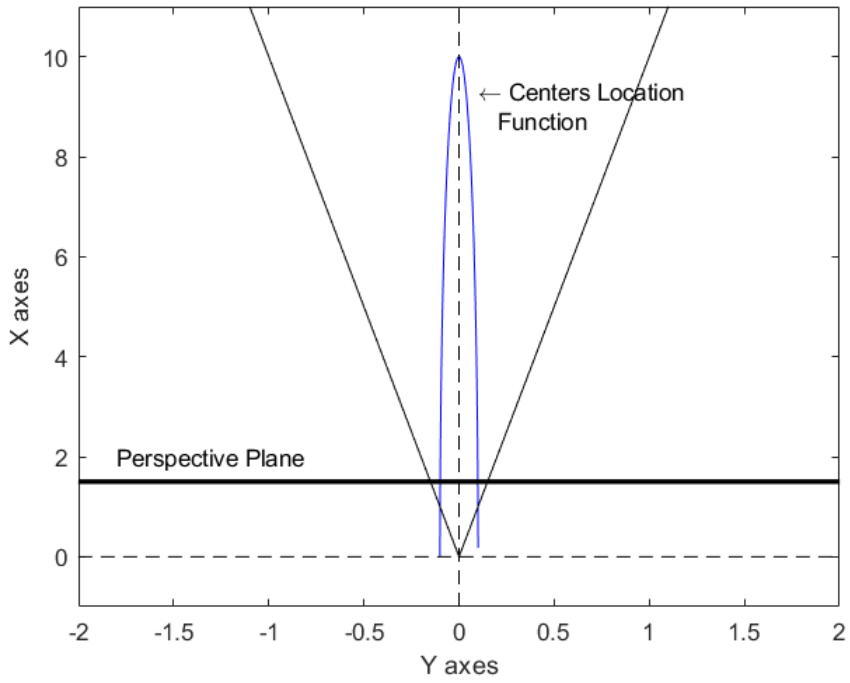


Figure 3.15: Centers Location Function

Proof. looking at figure 3.15, let $x_{center} = (x_D + x_E)/2$, and let G, which is not necessarily aligned with the focal axis, to represent the point of intersection of the segment $[DE]$ with the center line of projection, then x_D and x_E can be expressed as follows,

$$x_D = \frac{F \cdot y_D}{y_{D'}} \cdot DG \cdot \cos(\alpha)$$
$$x_E = -\frac{F \cdot y_E}{y_{E'}} \cdot EG \cdot \cos(\alpha)$$

where $y_{D'}$ and $y_{E'}$ are the Y-coordinates of the corresponding projections of y_D and y_E on the projective plane. Since the rotation is a mid-arc rotation $y_{D'} = y_{E'}$, this yields to the X-coordinate of the center of the segment as in eq. 3.19 The

same can be applied to determine the Y-coordinate of the center. Let $y_{center} = (y_D + y_E)/2$, substituting with y_D and y_E from the equations x_D and x_E which yield to 3.20. □

x_{center} and y_{center} here represent the coordinates of the center where α is the angle formed by the perspective plane and the line-segment $[AB]$.

We have shown in this section, in contrast to center segment rotation (Case(1)), mid-arc rotation cannot recover for the pose effect due to the uncertainty in extracting the tilted angle α. This bring us back to the need of using a third point on the segment, for example the center. But there is one case where two points are enough because in this case, the third one is always coincident with mid-arc. This is a unique case that agrees with the property of spherical targets, as we will see later. It is the case that complies to aspect (1), where the center is always located at the peak and x_{center} and y_{center} will be representing the position of a spherical target in space.

3.3.4 Projective Rotation from the Target Mid-Point of Projection

If now the rotation happens starting from the mid-point of the projection, it will be shown that this is useful to calculate eccentricity error, specially in the case of circular targets. The X-coordinate of the center, derived from the location curve, can be expressed by as in eq. 3.21.

$$x_c = \frac{F}{2} \cdot \left(\frac{DG}{y_{D'}} + \frac{EG}{y_{E'}} \right) \cdot \cos(\alpha) \tag{3.21}$$

Since $y_{D'}$ and $y_{E'}$ are different only one case is relevant here where $DG = EG$. This occurs when x_C is intersected with the centerline of projection, which in analogy with Aspect (2), this particular occurrence represents the case where the target lies in parallel to the image plane and can be deduced as it becomes equal to the angle of rotation performed.

3.3.5 Locating the Center from Three Points

We have shown in the previous section that two points are not enough to locate the center of the segment. We will show in this section the derivation of the full pose from three points.

We have derived previously the parametric equations that defines the centers location. Since the tilted angle α is not known, a third point is required to add one more constraint on the center location curve parametric equations. This equation is depicted in 3.22.

$$x_c + d\sin(\alpha) = \frac{F}{y_{d'}}\left[y_c + d\cos(\alpha)\right] \tag{3.22}$$

Proof. Considering a third point D located at distance d from the center C on the line segment and that we perform mid-arc rotation as shown in figure 3.16.

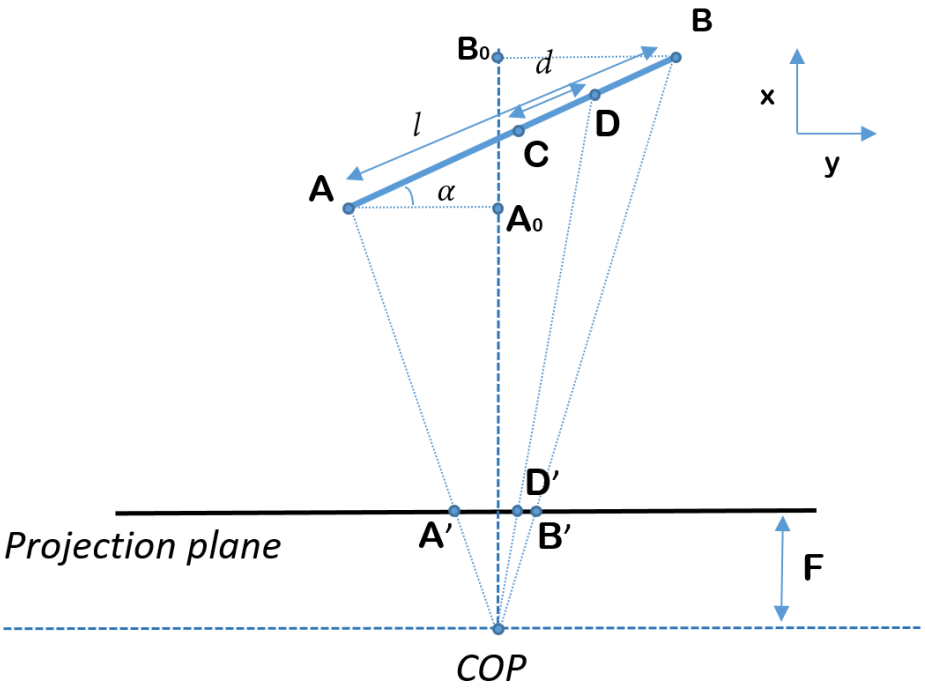


Figure 3.16: Three points - mid-arc projection

The equations that relate the point D to the center are illustrated below.

$$\begin{cases} x_d - x_c = d\cos(\alpha) \\ y_d - y_d = d\sin(\alpha) \end{cases} \implies \begin{cases} x_d = x_c + d\cos(\alpha) \\ y_d = y_d + d\sin(\alpha) \end{cases}$$

however, we can write the equation of the projected line from the point D as follow:

$$\begin{aligned} x_d &= a y_d \\ x_{d'} &= a y_{d'} \end{aligned} \implies a = \frac{F}{y_{d'}}$$

This lead to the additional constraint equation:

$$x_c + d\sin(\alpha) = \frac{F}{y_{d'}}\big[y_c + d\cos(\alpha)\big]$$

□

The system of equations become as in eq. 3.23

$$\begin{cases} x_c = \dfrac{L \cdot F}{2 \cdot y_{A'}} \cdot \cos(\alpha) \\ y_c = \dfrac{L \cdot y_{A'}}{2 \cdot F} \cdot \sin(\alpha) \\ x_c + d\sin(\alpha) = \dfrac{F}{y_{d'}}\big[y_c + d\cos(\alpha)\big] \end{cases} \tag{3.23}$$

From these equations we can extract the angle by replacing equation one and two into the third one. We get equation 3.24. Using eq. 3.24 we can recalculate the coordinates of the center x_c and y_c.

$$\alpha = \arctan\left(\frac{\frac{L \cdot F}{2 \cdot y_{A'}} - \frac{F}{y_{D'}} d}{\frac{F}{y_{D'}} \cdot \frac{L \cdot y_{A'}}{2 \cdot F} - d}\right) \qquad \alpha \in [-\tfrac{\pi}{2} \quad \tfrac{\pi}{2}] \tag{3.24}$$

Knowing the coordinates of the center and the tilted angle of the segment we can

recover the full pose in the camera frame by replacing in eq. 3.25.

$$\xi_{(x_c,y_c,\alpha)} = \begin{bmatrix} \Omega & \tau \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & x_c \\ \sin(\alpha) & \cos(\alpha) & y_c \\ 0 & 1 & \end{bmatrix} \tag{3.25}$$

3.4 Foreshortening Factor and Correction Methods

Previously, we explored the pose and distance factor. In this section, we will look at the foreshortening factor. This time the line-segment is considered to be away from the focal axis. Because of this, the line is subject to foreshortening distortion.

Like before, we define the distortion factor to be the the ratio of the projected length over the real length of the segment and we denote this factor by the small μ as in eq. 3.26, but this time we append the letter T to distinguish between the total distortion factor and what we calculated before. Noting here that μ_T belongs to $[0, 1]$.

$$\mu_T = \frac{|\, y_{B'} - y_{A'}|}{L} \qquad \mu_T \in [0 \quad 1] \tag{3.26}$$

$|y_{A'} - y_{B'}|$ is the projected length and it can also be written as function of the projected length after rotation $|y_{D'} - y_{E'}|$ as in equation eq. 3.27.

This leads to the equation 3.28 where μ_T can be written in terms of pose and distance factor, and an additional factor that we call here foreshortening distortion factor (FDP). We denote the last factor by μ_3. Finally, we can put all the distortion factors in one equation, as in eq. 3.29.

$$\underbrace{|y_{A'} - y_{B'}|}_{\text{Projected Length}} = \underbrace{[y_{D'} - y_{E'}]}_{\substack{\text{Projected Length before} \\ \text{foreshortening distortion}}} \cdot \frac{[F^2 + \frac{y_{C'} \cdot y_{A'}}{2}] \cdot [F^2 + \frac{y_{C'} \cdot y_{B'}}{2}]}{F^2 + y_{C'}^2} \tag{3.27}$$

$$\underbrace{\mu_T}_{\substack{\text{Total distortion}\\ \text{factor}}} = \underbrace{\mu}_{\substack{\text{Distortion factor}\\ \text{from pose and distance}}} \cdot \underbrace{\frac{\left[F^2 + \frac{y_{C'} \cdot y_{A'}}{2}\right] \cdot \left[F^2 + \frac{y_{C'} \cdot y_{B'}}{2}\right]}{F^2 + y_{C'}^2}}_{\text{Foreshortening distortion factor}} \tag{3.28}$$

$$\underbrace{\mu_T}_{\text{TDF}} = \underbrace{\mu_1}_{\text{DDF}} \cdot \overbrace{\mu_2}^{\text{PDF}} \cdot \underbrace{\mu_3}_{\text{FDF}} \tag{3.29}$$

To sum up, in this section we presented a detailed analysis of how distortion is being propagated and come up with a model of the total distortion on a line-segment from perspective projection. Since a line-segment form the basis of every structure, it was very important to understand and model these distortions before proceeding in building localisation concept. We were also able to show that mostly these distortions could be eliminated, based on the cases discussed.

Now that we have acquired all these proofs and information, we can start building our first marker to extract our position. We can start thinking about two possible use cases for using our segment marker to localising our vehicle. From what we developed, before we can conclude that we need at least three points on the line to extract the relative distance and the pose of the line without also forgetting our singularity point, where the projection becomes a point when the segment is co-linear with focal axis. Or we can use two points on a line but the tilted angle of the segment should be fixed relative to the image plane.

Use case (1): We have a camera mounted on a moving vehicle in a small room. The camera is looking at three co-linear points placed in the room with known length where we would like to extract the camera relative distance while moving on the 2D surface.

Use case (2): We have a mounted camera on a moving vehicle in a small room. The camera is looking towards the ceiling with fixed angle where two point features are fixed and can be detected by the camera. We would like here to extract the camera relative distance moving on a 2D space.

Since it is just a line segment, it is very likely to fall into a singularity. In this case the system can return infinite solution. Another restriction is that only distance

can be extracted but not the position. And this is due to a major fact of the co-linearity of our marker. Co-linearity provokes the system to lose a degree of freedom. For use case (1), we can consider using a three points segment approach while for case (2), we can use the approach of detecting the edges of the segment and since the camera angle is fixed and is moving on a 2D surface, extracting the distance using this approach becomes valid.

In the next section we extended the work for circular and spherical targets. These kind of targets have there own usage advantage as we will see later.

4 Circular and Spherical Targets

4.1 Introduction to Spherical and Circular Targets

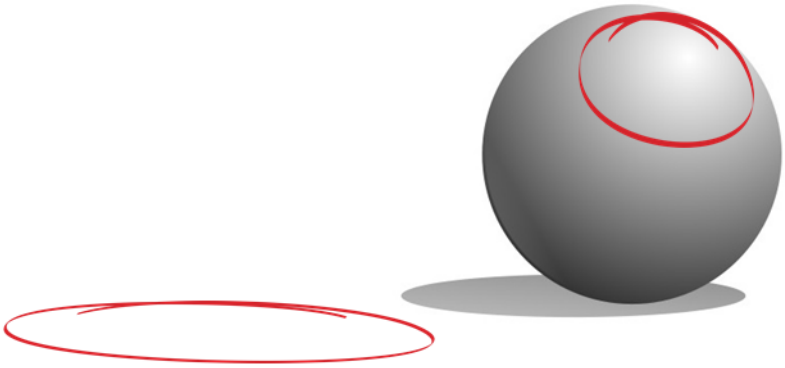


Figure 3.17: Spherical targets

At the beginning of this chapter, we presented pose, distance and foreshortening effect on a line segment, followed by a demonstration on how distortion effect can be eliminated. We place the basic building blocks towards building a complete marker-based concept that can capture position and orientation. We now extend the work and development to circular and spherical targets, which play a more realistic role as markers.

Circular and spherical targets are commonly very employed in Photogrammetry and many different industrial applications, specifically for optical measurements

(57). Circular targets in particular, are mostly used in close range Photogrammetry for high accurate measurements. It has been commonly used as artificial markers in machine vision and has been payed continuous attention for a decades of historical usage (140). While circular targets have their own specificity, spherical targets have scored some additional advantages in different applications where planar targets are not well suited. This kind of target is mostly used for calibration processes like in LIDAR, LASER and other 3D sensors calibration (106), (20).

In contrast to other types of targets like checkerboard targets, circular and spherical targets offer numerous different advantages (81). Some crucial advantages of using circular and spherical targets are:

- The symmetry in the pattern
- Invariant in scale and invariant in rotation aspect
- The existence of many algorithms for detection and measurement of the target in images (e.g., centroid and contour-based methods)
- Finally, the size of the target can be very small and still be functional and efficient for the measurements

However these kind of targets often present major drawbacks as well. For example:

- The position of the light source and the material used can impose a lot of reflection which affect the detection and measurement of the target
- Other bright blobs in the image can be easily mistaken with circular and spherical targets
- Eccentricity Error, which has direct impact on position of the target in the image

In addition, the use of spherical targets presents even more advantages than using circular targets, like a better visibility from all different angles and a disadvantage sometimes in the measurement of the accuracy coming from the blurred edges due to the curvature of the sphere.

4.2 Projection of Spherical Targets

Before proceeding in exploring the eccentricity problem originated from the perspective distortion, we would like to evaluate with a short introduction how spheres are projected.

Some properties for projection of a sphere can be stated here. If the sphere is placed anywhere in front of the projective plane, its projection preserves its conic property. Furthermore, and because of its symmetrical aspect, the silhouette of a sphere, seen from any angle of view, represents a circle before being projected. This implies, there is always a circular cross-section of that sphere that is projected to the projective plane, regardless of the sphere position and orientation, but its projection is not necessarily a perfect circle.

This cross-section 3.18 before projection lies always perpendicular to the line crossing the mid-arc of the projection. Figure 3.19 depicts these properties.

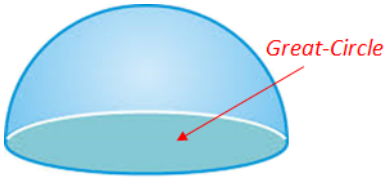


Figure 3.18: Great-Circle formed by the cross-section

From the perspective view and due to the added distortion effects, the projection of the circular cross-section becomes egg shaped. And because the line of projection at the edges is tangential to the sphere, it is important to note that the projected cross section is slightly in front and parallel to the great cross section as shown in figure 3.20.

To derive the relationship of the cross-section projection and the position of the sphere in space, we perform a mid-arc projective rotation towards the center, which provokes always the cross-section to become parallel to the image plane and eases its calculation. On the other side, this also tells us that the initial tilted angle, of the cross-section for the image plane, is equal to the angle of rotation

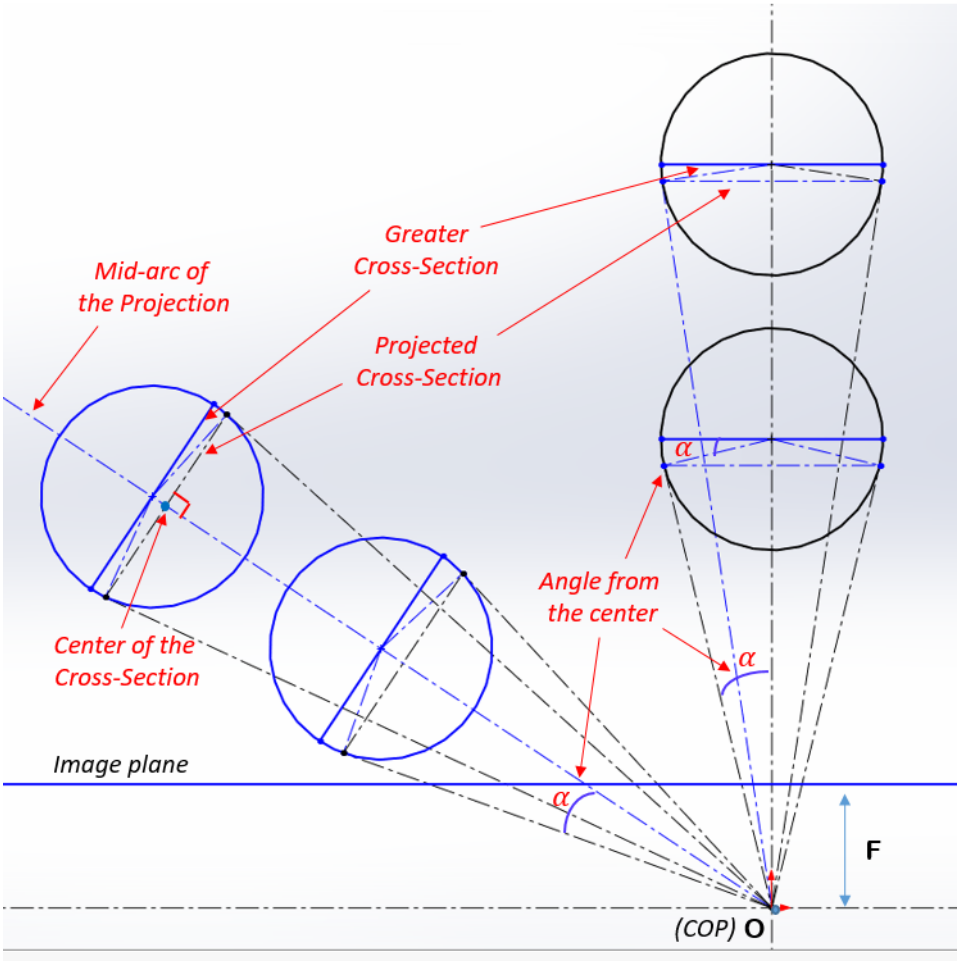


Figure 3.19: Sphere seen from the top-view of the projection. Spherical Cross-section projected to the image plane.

performed. The cross section projection becomes:

$$y_{D'_T} = \frac{F \cdot L \cdot \cos(\alpha)}{2 \cdot x_{A_T}} \tag{3.30}$$

$$y_{E'_T} = -\frac{F \cdot L \cdot \cos(\alpha)}{2 \cdot x_{A_T}} \tag{3.31}$$

We note here that α represents the angle formed by projected point $O\vec{D}'_T$ to the center of the image plane $\vec{OI}$, as shown in the figure 3.21.

Proof. Looking at the triangle D_TOG in figure 3.21, depicted in red, and calculating the sum of its angles yield $\alpha + \alpha_1 + \frac{\pi}{2} = \pi$ with $\alpha_1 = \frac{\pi}{2} - \alpha_2$ implies $\alpha = \alpha_2$.

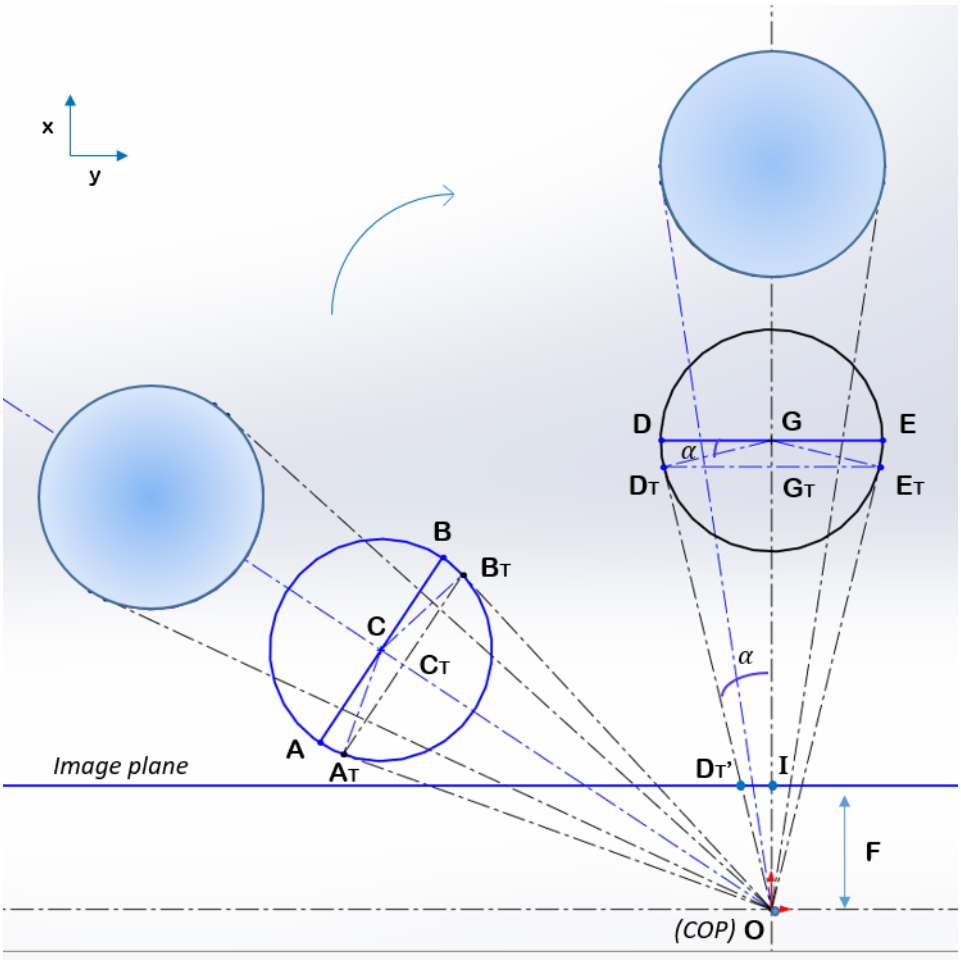


Figure 3.20: Projection of the sphere from its tangents lines

Since $O\vec{D}'_T$ is tangent to the sphere, $= DG = \frac{L}{2}$ (L is the radius of the sphere) and $D_T G_T = DG \cdot \cos\alpha$ implies,

$$\begin{aligned} y_{D'_T} &= \frac{F \cdot D_T G_T}{x_{A_T}} \\ &= \frac{F \cdot L \cdot \cos(\alpha)}{2 \cdot x_{A_T}} \end{aligned}$$

□

In this section we model the spherical projection. This model will be used whenever we have a projected sphere in the image, with no exception.

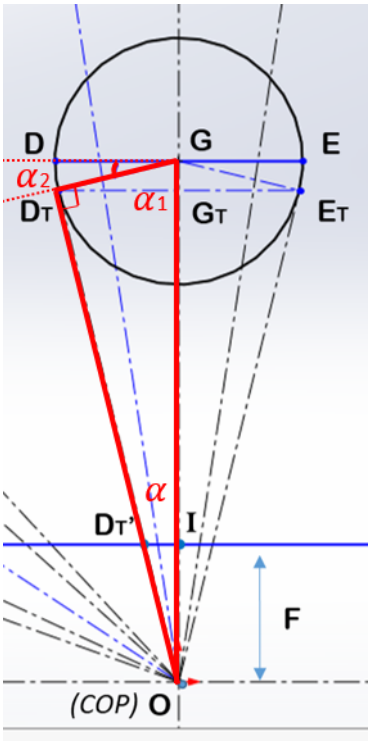


Figure 3.21: Angle formed by the cross-section

4.3 Eccentricity Caused by Distortion Effect

As discussed in the previous section, one of the main drawback in using circular and spherical objects as a target in images is Eccentricity.

A concise definition in geometry of Eccentricity is that Eccentricity measures how nearly circular is an ellipse. It is a ratio describing the shape of a conic section, for example, circles which are particular of ellipses but with zero eccentricity. Perspective Eccentricity is due to one of the properties discussed previously in perspective geometry, which is the distortion effect, mainly the pose and foreshortening effect. The reason is that perspective projection scales all the points according to the distance to its projection on the view axis and not according to the actual distance to the center of projection. Distance effect has no impact on eccentricity.

Circles and spheres are projected into an ellipse on the image plane in perspective

projection. This provokes the projected center of the circle or the sphere to be displaced into a new center of an ellipse creating eccentricity. Figure 3.22 depicts the formation of Perspective Eccentricity denoted by e. It shows that the center of an ellipse formed by the projection of a circle or a sphere is deviated from the real center projection onto the image plane (81). Since the center of the ellipse is very crucial for accurate measurement of the distance to the original target for example, it is of high importance to calculate the displacement of the true center (81).

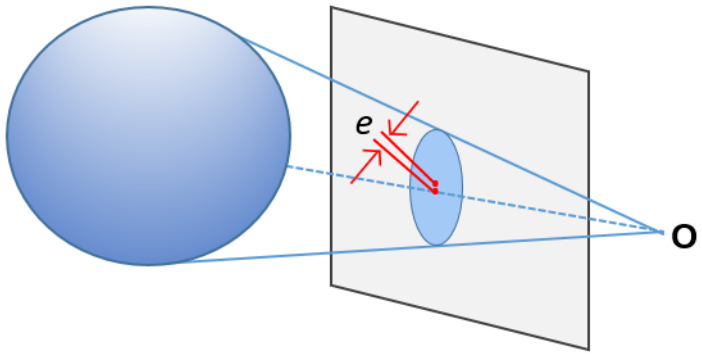


Figure 3.22: Eccentricity of the projection of a sphere

4.3.1 Perspective Eccentricity of Imaged Circle

In this section we will start by exploring eccentricity on circular targets.

From perspective projection, the image of a circle is mapped to an ellipse where the center of the elliptical image is not identical to the true projected center of the target, except when the circle and image are in a parallel position to each other. This phenomenon is known by eccentricity. Eccentricity of circular target was first raised by Lenz and Fritsch in 1988, and later on considered by Beyer and Riechmann in 1992 to be the source of systematic geometric measurement error and which was proven later to be true. Many researches later were being conducted to derive solution for correction on the images. Since the normal of the target surface is not known, solution becomes more advanced and complex. Motivated by this topic, we developed a unique, efficient and robust method for fixing eccentricity.

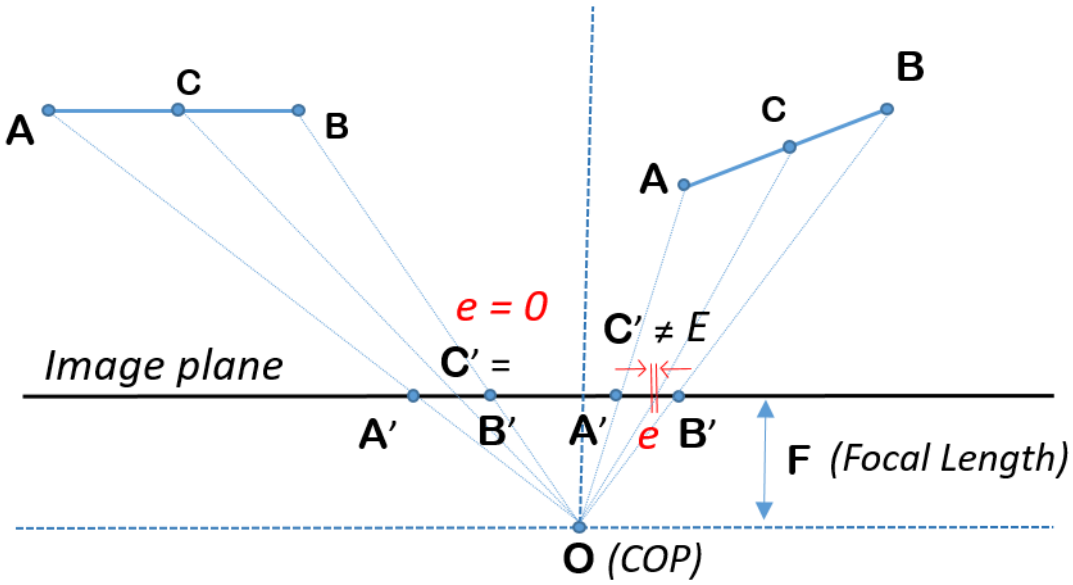


Figure 3.23: Eccentricity of a circular target projection

To calculate eccentricity, we need to initially locate the center of the projected ellipse before proceeding in finding the true center. Practically, the center of the ellipse is determined by means of digital processing. Some ways to compute the actual center of a circular target in digital images are by deriving the center-of-gravity or pixel-centroid operation on particular windows in the image 3.32, correlating with a reference pattern or by using the outline methods for circles or ellipses. Some of these methods are depicted in the book of Close-Range Photogrammetry and 3D Imaging (134).

$$X_M = \frac{\sum_{i=1}^{n} x_i \cdot T \cdot g_i}{\sum_{i=1}^{n} T \cdot g_i} \qquad Y_M = \frac{\sum_{i=1}^{n} y_i \cdot T \cdot g_i}{\sum_{i=1}^{n} T \cdot g_i} \qquad T = \begin{cases} 0 & \text{for } g < t \\ 1 & \text{for g} \geqslant \text{t} \end{cases} \tag{3.32}$$

Calculating the actual center is important for the calculation of the eccentric error. Now that we have center of the ellipse, we can proceed with its derivation and elimination. From literature, there are few different approaches to estimate the correct center. Matsuoka, et al, focused on a particular case where the circle lies on the horizontal plane. Matsuoka, et al. (2016) published a numerical general

formula for calculating eccentricity from different parameters like the target size, position in image, camera attitude and focal length. While Luhmann et al. (2014) described a way for evaluating the effect of eccentricity and gave an estimation of the ellipse parameters from contours points from Dolds equation in 3.33.

$$e' = r_m - \frac{c}{2}\left(\frac{R_m + \frac{d}{2}\cdot\sin(90-\alpha)}{Z_m - \frac{d}{2}\cdot\cos(90-\alpha)} + \frac{R_m - \frac{d}{2}\cdot\sin(90-\alpha)}{Z_m + \frac{d}{2}\cdot\cos(90-\alpha)}\right) \tag{3.33}$$

- *e': eccentricity*
- *d: Target diameter in object space*
- r_m*: image radius of the projected target*
- α*: angle formed by image plane and target*
- R_m*: latteral offset of the target to the optical axis*
- Z_m*: Object distance from vanishing plane*
- *c: principal distance*

Eq. 3.33 shows that eccentricity depends on the radius and the principal distance, and most importantly on the angle α between the image plane and the target. According to Dolds equation 3.33, if α is zero, eccentricity becomes zero. An important notice here, that this is in accordance and in analogy with Aspect (2) described in section 3.3.1, which means that this occurs when the circle plane and the image plane are strictly in parallel. These cases are illustrated in the figure 3.23. Seen from the top, the flat circle target appear like a thin line and can be represented by the segment $[AB]$ in the world coordinate system. The right segment $[AB]$ has zero eccentricity while the left one, and since its angle α is not zero, its center is deviated and thus, its eccentricity is different than zero.

Based on these facts, we proposed here a model for calculating the relative distance and pose of circular and spherical target. Before proceeding we would like first to add some building block into our workflow. The updated workflow is presented in 3.24, which gives a visual explanation of the proposed model.

The goal again is to acquire the distance and pose of our circular target, considering that we have a circular target lying on a flat surface and placed anywhere in

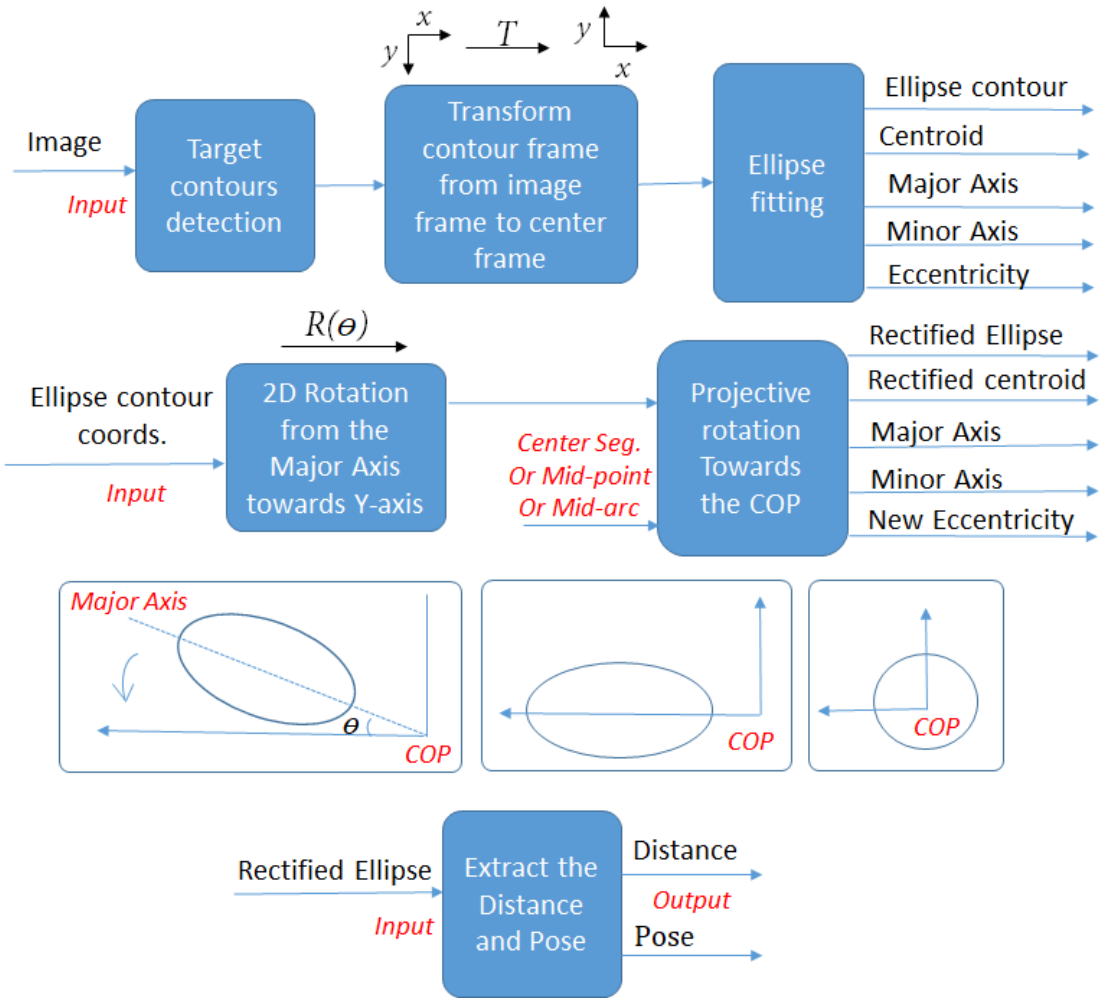


Figure 3.24: Workflow

space within the field of view of a pinhole camera. The circle is projected into the image frame in form of an ellipse. We present below every stage of the workflow in detail.

(1) In step one, we process the image and detect our target.

(2) Since we would like to use this model for simulation, we consider transformation between image frame and the center frame. The transformation T used is a simple rotation and translation, as in eq. 3.34, where imW and imH are the image width and height respectively. We use this transformation to transform all the contour points from the image to the center frame. The new contour coordinates are denoted by $E' = T \cdot E$.

(3) In the third stage, we fit our data into an ellipse from which we extract the image centroid, the major and minor axis, and perform an evaluation of the current eccentricity.

(4) Stage four is the beginning of the correction phase. A rotation of all the ellipse points is performed to align the major axis with one of the coordinates axis. In our case we use the Y-axis.

(5) This is the final stage of correction. In this stage we perform a mid-point projective rotation to all the points. This means that every point is mapped to its corresponding point as if the ellipse lies at the center. The mapping is made by applying eq. 3.12

(6) The workflow ends, as before by extracting the pose and distance.

$$T = \begin{bmatrix} 1 & 0 & 0 & \frac{imW}{2} \\ 0 & cos(\pi) & -sin(\pi) & \frac{imH}{2} \\ 0 & sin(\pi) & cos(\pi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \frac{imW}{2} \\ 0 & -1 & 0 & \frac{imH}{2} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

In this section we evaluated circular target and make use to design the workflow for a correction algorithm that we will use next in the simulation. For circular

target, we perform rotation towards the center, which is effective, but in all cases the tilted angle should be known. Since our circle is foreshortened to a line, the previous developed study and correction method of line segments can be applied here.

In the coming section we put light on spherical targets.

4.3.2 Perspective Eccentricity of Imaged Sphere

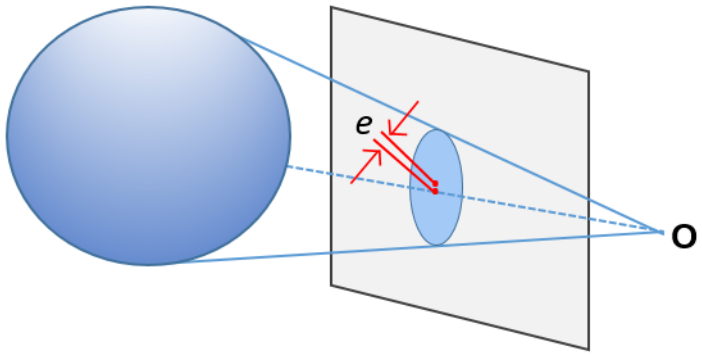


Figure 3.25: Eccentricity of the projection of a sphere

In this section, we use spheres as targets. Spherical targets have their own advantages on circular targets because of their natural 3D shape. The 3D symmetrical shape of the sphere makes its projection independent of its pose. This means that we don't need to know and to take care of the pose in order to generate the distance. The only parameter that needs to be corrected is its eccentricity. Furthermore, in order to have no deviation of the projected center when the target is spherical, the position of the center of the sphere should coincide with the view axis. Because there is always a circular cross section that is projected to the image plane and is perpendicular to the mid-arc, we can simply perform a mid-arc projective rotation towards the COP. This property makes spherical target agrees with Aspect (1) since its center is always coincident with mid-arc axis. Eliminating distortions can be simply achieved by performing a mid-arc projective rotation towards the center.

We use the same approach and workflow for spheres as in 3.24. This time we run a short simulation of different positions of spheres in the image. Figure 3.26 shows the initial position of the sphere. We can clearly see as well that the further away from the center the more the impact of perspective distortion is on the targets.

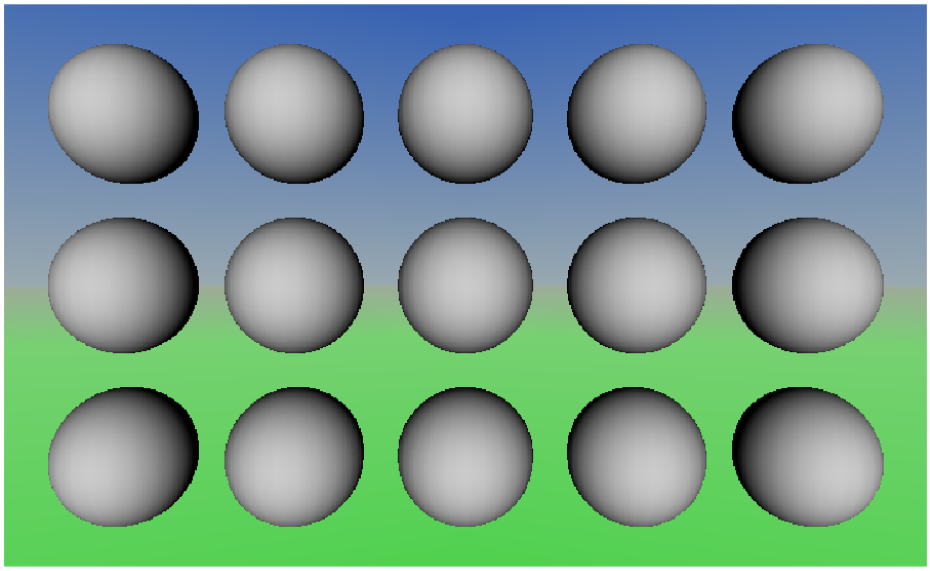


Figure 3.26: Spheres positioned in within the field of view of the camera

Stage one is illustrated in figure 3.27, where we detect contour and extract the centers. In figure 3.28 we change frame and fit ellipses to the detected contours; these processes are embedded in stage two and three. From the fitted ellipses, we extract the inclination angle of the major axis and use the angles to perform a 2D rotation so that the new major axis of the ellipse is coincident with the Y-axis. Figure 3.29 shows the result of stage five after mid-arc projective rotation toward the COP. It is clear from the new circular anatomy of the projections that the method has efficiently restored the original shapes of the spheres and made them free from pose and foreshortening effect. In this stage as well we extract our new eccentricity and compared them with the original eccentricity.

The results are shown in table 3.1. Our evaluation to the results is stated here:

- first, eccentricity has been drastically reduced independently from the position of the spheres within the image.

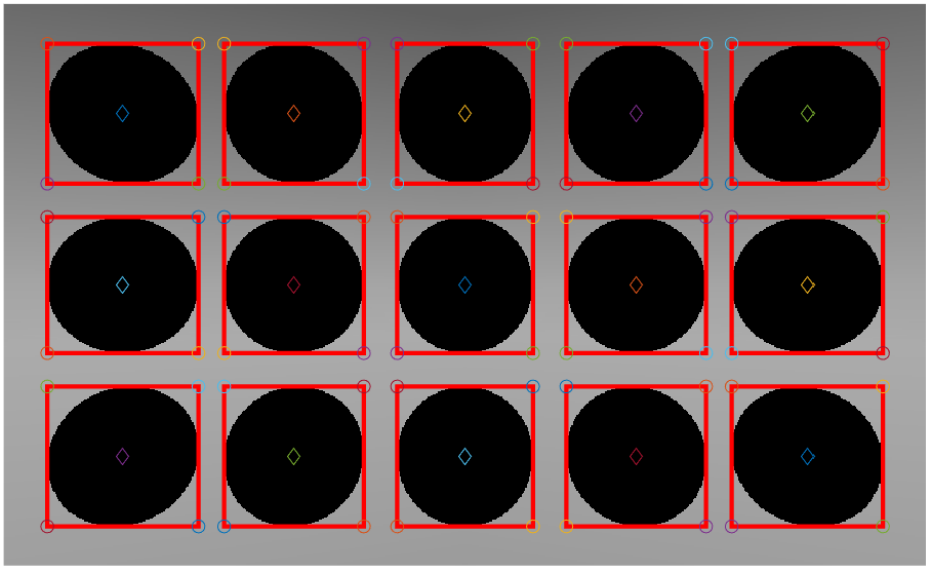


Figure 3.27: Contour and centroid detection

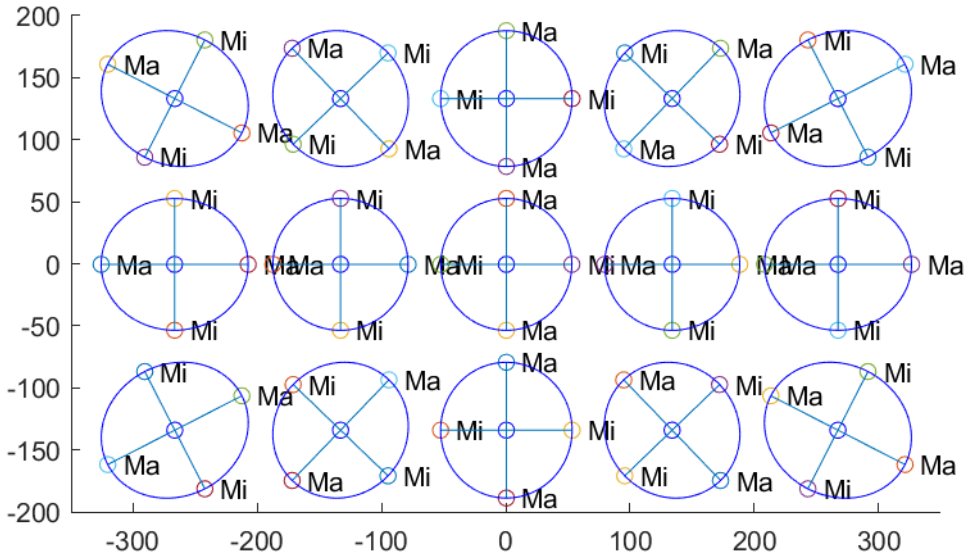


Figure 3.28: Ellipse fitting to the contours and extraction of minor and major axis

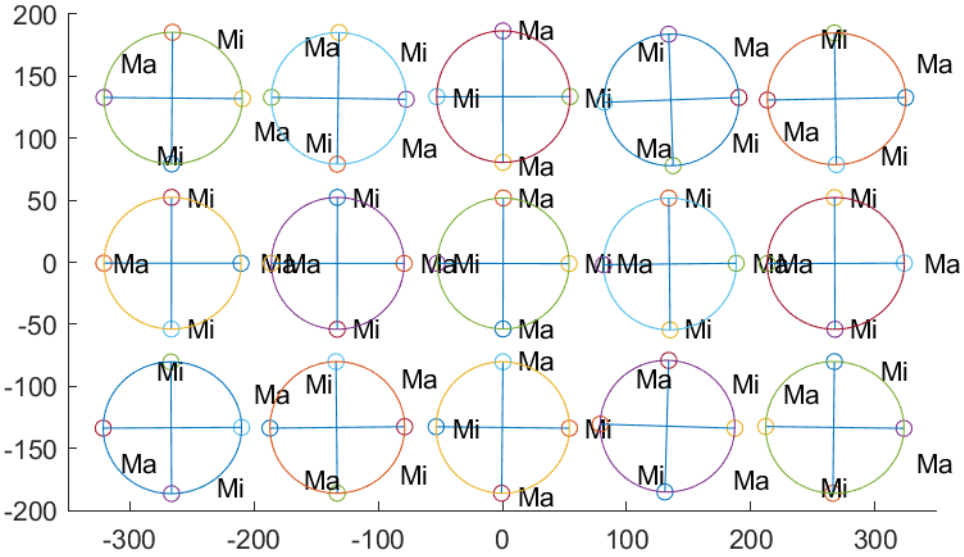


Figure 3.29: Rectified ellipses in the image

- Second, a value below 0.2 is considered to be nearly a perfect circle. And since we are dealing with digitized images with discrete value, noise can be induced on contours which also affects the ellipse fitting process.

- Third, actual centers of the detected contours have also some induced noise as we can see in the data provided in table 3.1.

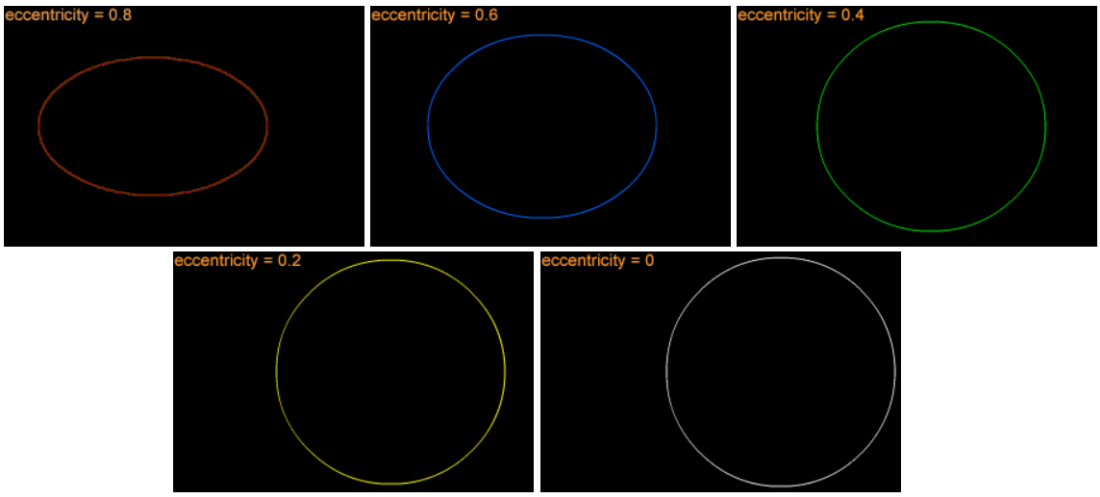


Figure 3.30: Eccentricity Evaluation

In conclusion, since the center can be always determined, eccentricity can be always calculated and the distance to the center can also be derived, which can

Centroid X	Centroid Y	Ecc. (Original)	Ecc after correction
-266.4836	133.0014	0.4859	0.1897
-266.6037	-0.5000	0.4433	0.1525
-266.4836	-134.0014	0.4859	0.1941
-132.9943	132.9979	0.3331	0.1194
-133.0370	-0.5000	0.2181	0.0859
-132.9943	-133.9979	0.3331	0.1048
0.5000	132.9945	0.2473	0.1006
0.5000	-0.5000	0.1154	0.0875
0.5000	-133.9945	0.2473	0.1323
133.9943	132.9979	0.3331	0.2033
134.0370	-0.5000	0.2181	0.2105
133.9943	-133.9979	0.3331	0.1093
267.4836	133.0014	0.4859	0.2072
267.6037	-0.5000	0.4433	0.2060
267.4836	-134.0014	0.4859	0.2531

Table 3.1: Effect of perspective correction of Eccentricity.

be used to extract the distance to the target and the 3D relative position. Here we should mention and distinguish two cases. The 3D relative position of the target can be calculated if we consider the object is moving withing the field of view of the camera and that we know the original size of the target. The approach is not reversible. Determining the camera relative position if the sphere is fixed, is not possible. If we have a fixed spherical marker and a camera looking at the target, only the distance can be derived. The reason is that natural symmetrical shape of a sphere does not provide any additional information about its pose, and obviously, this sort of marker only provides information about one point in space, which is not enough for 3D pose of the camera estimation.

5 Summary and Discussion

In this chapter we covered the basic elements for building our marker based localisation concept. We went deeply into every aspect of perspective distortion and provided ways of eliminating these kinds of distortions. We covered advantages and immaturity of such marker and what we concluded is that both line markers and spherical markers are not mature geometries when standing along, and when it comes to camera pose estimation with respect to the target. Geometry should be combined. In the next chapter we expand all the work and explore a new marker geometry that can hold for more information relevant for camera position and orientation.

Chapter 4

Geometry Model for Marker-Based Positioning

Contents

1 Chapter Summary

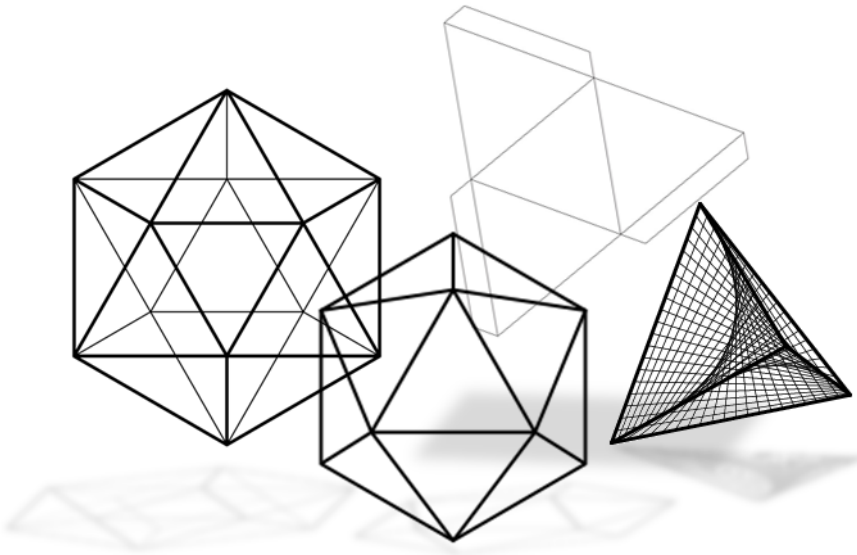


Figure 4.1: Markers Geometry Modeling

Having the perspective projection of three points in $3D$ space, which constitute the vertices of a predefined triangle size, it is possible to extract the position of each of the vertices (55). Fishler and Bolles were one of the first to debut immersion on this problem (46), referring to it as *Perspective-n-Points (PnP)*.

> Given the relative spatial locations of n control points, and given the angle to every pair of control points from an additional point called the Center of Perspective CP, find the lengths of the line segments "legs"joining the CP to each of the control points. We call this the "perspective-n-point problem"(PnP).

For more than decades, this problem has been studied in Photogrammetry and later on, in the field of computer graphics and Computer Vision, for trying to estimate the position and orientation of a calibrated camera from image points. It became the fundamental problem in the field of Computer Vision, Computer Graphics, Augmented Reality and Virtual Reality, and of many other applications like mobile robotics and navigation.

Space Resection is another terminology which was initially employed and became widely used afterwards, to describe relationship of the camera pose in terms of position and orientation from projected $2D$ n-points in the image, corresponding to $3D$ n-points in the scene. A great definition was presented by Moffit Mikhail in 1980:

> The term *Space Resection* is the name given to the process in which the spatial position and orientation of photograph is determined, based on photogrammetric measurements of the images of ground control points appearing on the photograph.

There are many approaches that try to come up with solutions to the PnP and *Space Resection* problem. Literatures are many (80), (52), (118) and are more explored in the earlier chapter 2. While all these methods mainly focus on control points, we rather look at the problem from different perspectives. We think as, a geometry as a whole rather than individual control points spread in space. We often feature shapes that perfectly fit and correspond to each other, and look at patterns that seem uncommonly accurate and precise.

We are now at the point where we can start building a realistic geometry model that can capture relevant information related to pose estimation. We will use the fundamental knowledge and building blocks from the former chapter to design our marker with novelty. We then make use of the previous techniques for eliminating perspective distortion for the purpose of achieving a high accurate marker-based localisation model.

If we consider a spherical marker, as in chapter 3, we notice that due to its homogeneous surface and symmetrical shape, pose of the sphere in the sense of orientation in the image can never be recovered. But if we consider now that the sphere is re-sectioned into n parts, without disassembling its spherical shape, but in fact letting every part to be unique and distinct by marking it with different color, recovery becomes possible. Now that we rotate the sphere around its center, its orientation changes, and we can still be able to track its direction in which the sphere is being oriented. Furthermore, if we assign to every part a center,

every center will have relative $3D$ position with respect to the original center of the sphere. Imaging now that we carve the space between these centers. what is left of the sphere are point clouds located on spherical surface. If we label these centers in such a way to locate their projections on the image plane, meaning that we extract the $3D$ to $2D$ correspondences in the image, we can then locate our sphere in space, extract its orientation, extract the camera relative position and orientation with high precision. In computer vision and related themes, the terms position and orientation are linked together and concatenated into one word which is "pose". We will use the word pose from now on to refer to both terms together. In this chapter we design a spherical marker with n points located on its surface for the purpose of creating robust features for pose estimation. We will start with $n = 3$, which is the minimum number for which we get a defined number of solutions as claimed by Fishler and Bolles (46), (55), and try to fit best the points on a sphere.

2 Approach and Concept

In this section we will introduce the reader to the approach we follow to come up with the solution for our perspective pose estimation. We will shortly start by drawing the diagram in figure 4.2 depicting the workflow and algorithm behind our approach.

In our approach, we try to find a transformation that maps perspective projection into parallel projection. In parallel projection, objects in 3D space are projected onto a fixed plane called projective plane. Parallel projection is good for exact measurements which is in contrast to perspective projection, where measurements and sizes vary as the object moves away from the image plane. One common property to both projections is that both do not preserve angles but when it comes to parallel projection, lines remain parallel. A major characteristic behind these differences is that the COP in parallel projection is considered to be infinitely far from the image plane. Because we only have access to one of the three Euclidean planes, depth is also lost.

Combining both projections is one of the key to restore depth. We need the exact

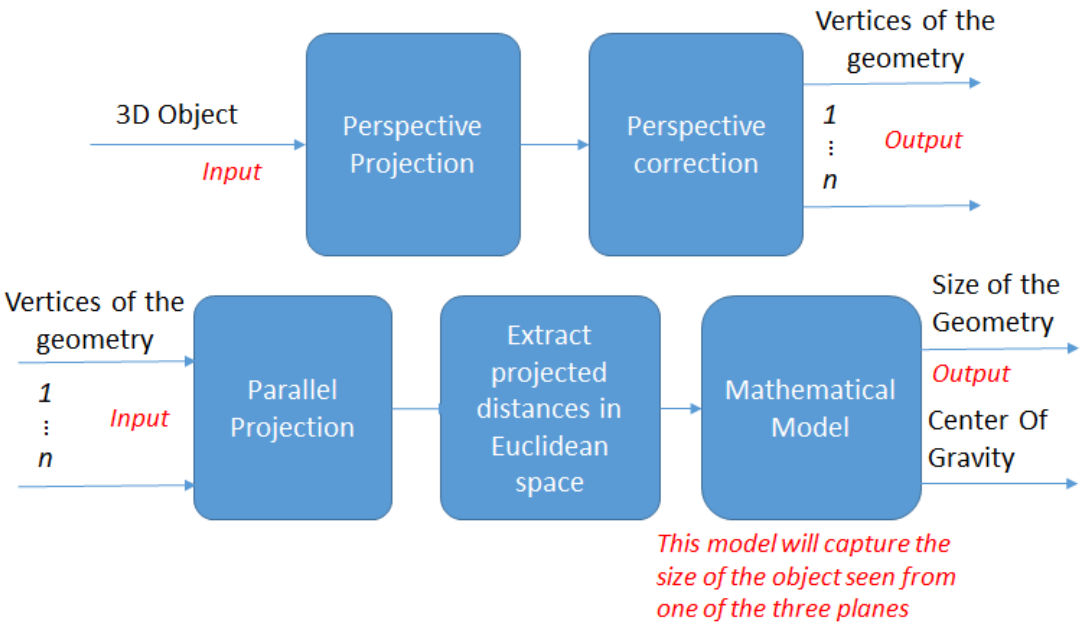


Figure 4.2: Workflow

measurements of parallel projection and the object size variation from perspective projection. For this reason, we first use our image from the perspective, apply some perspective correction to vertices then input these vertices into the parallel projection model.

In the next sections, we will illustrate the mathematical model we designed to capture the vertices and output the corresponding size of the object. We create a virtual spherical point clouds with a minimum n points require for a minimum number of solutions.

3 Parallel Projection of Equilateral Distances

This section is intended to explore the mathematical approach we developed to come up with the model that capture a number of vertices and output the corresponding object size. It turned out that for any three points in space we can associate a sphere onto which these points are embodied. Another aspect is that for $n = 3$, which is the minimum number of points, the radius of the sphere can still be calculated from the euclidean distances when projection onto the image

plane is parallel independently from the position of the vertices in space. We will call this model the virtual sphere model (16).

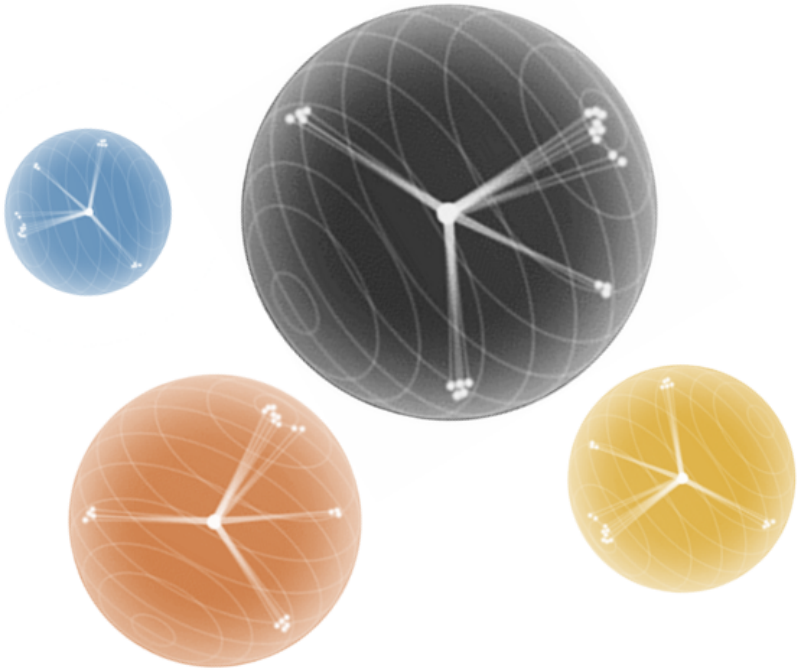


Figure 4.3: Virtual Spheres features

To prove our theory, we start with the most simple distribution of the three points. We will assume that the points are equidistant from each other forming the simplest equilateral geometry.

3.1 Representation in Spherical Coordinates

Considering that the marker features are embodied on a spherical surface, we can use spherical coordinates to represent its vertices in space. If we let A, B and C represent the vertices of an equilateral triangle, as in figure 4.5, we can define these vertices in spherical coordinates (ρ, θ, ϕ) system with the convention described in figure 4.4, using the following coordinates equation 4.1:

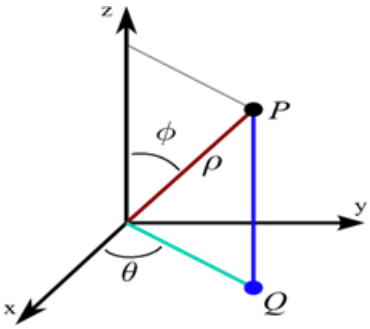


Figure 4.4: Spherical coordinate System.

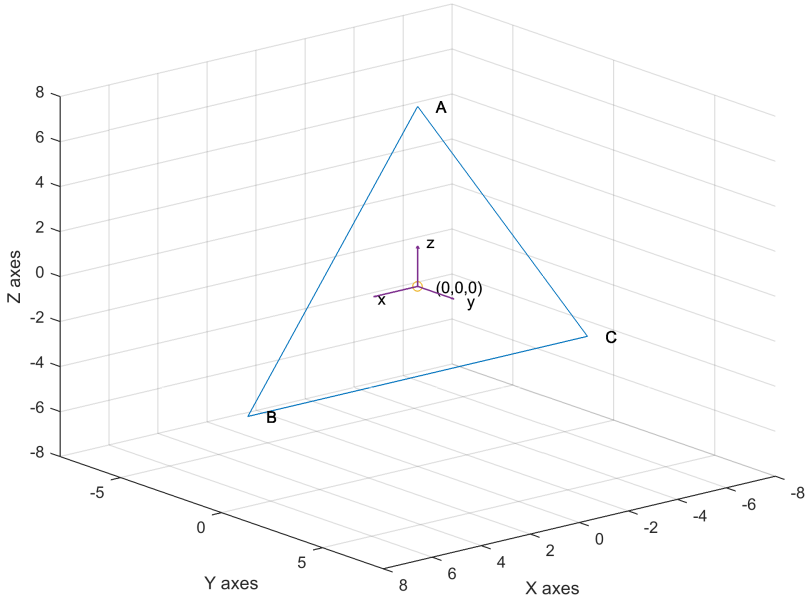


Figure 4.5: Equilateral Triangle in spherical coordinates.

$$A = \begin{bmatrix} \rho\sin(\phi)\cos(\theta) + x_G \\ \rho\sin(\phi)\sin(\theta) + y_G \\ \rho\cos(\phi) + z_G \end{bmatrix}$$

$$B = \begin{bmatrix} \rho\sin(\phi + \frac{2}{3}\pi)\cos(\theta) + x_G \\ \rho\sin(\phi + \frac{2}{3}\pi)\sin(\theta) + y_G \\ \rho\cos(\phi + \frac{2}{3}\pi) + z_G \end{bmatrix} \qquad \begin{matrix} \rho \in [0,\ \infty] \\ \theta \in [0,\ 2\pi] \\ \phi \in [0,\ \pi] \end{matrix} \tag{4.1}$$

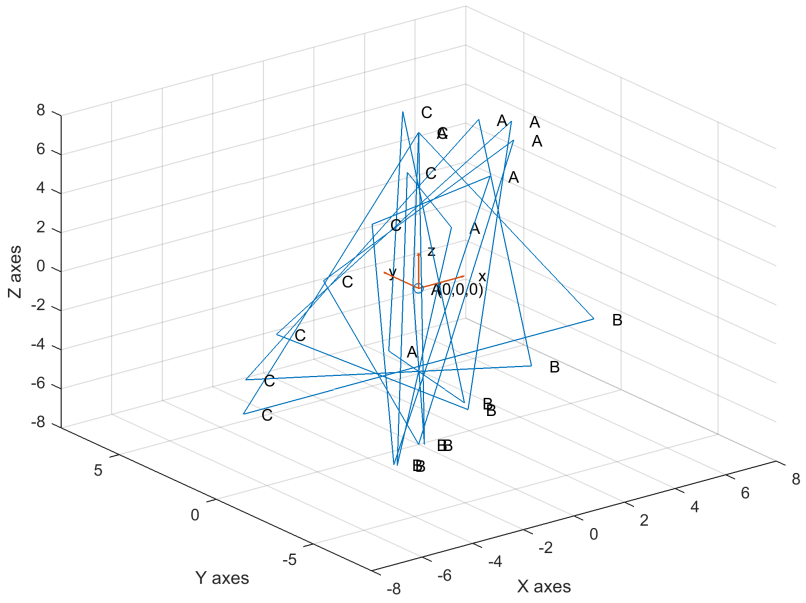
$$C = \begin{bmatrix} \rho\sin(\phi - \frac{2}{3}\pi)\cos(\theta) + x_G \\ \rho\sin(\phi - \frac{2}{3}\pi)\sin(\theta) + y_G \\ \rho\cos(\phi - \frac{2}{3}\pi) + z_G \end{bmatrix}$$

where ρ is radius of the sphere, θ and ϕ are the azimuthal and the polar angle respectively, and x_G, y_G and z_G are the coordinates of its center. Defining our system in spherical coordinates allows us to orient the triangular marker in all the direction by varying θ and ϕ as in figure 4.6.

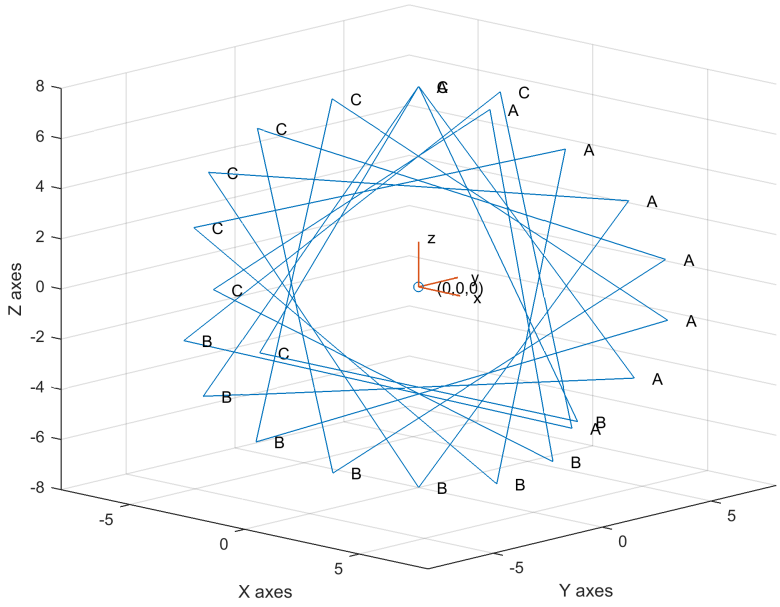
3.2 Parallel Projected Distances in Spherical Coordinates

From the coordinates equations of A, B and C we extract the parallel projected distances between all the three vertices in every fixed Euclidean plane using Euclidean distance. Equations 4.2, 4.3 and 4.4 represent the projected distances in the plane (YZ), (XY) and (ZX) respectively.

$$\begin{aligned} AB_{YZ} &= \sqrt{(y_A - y_B)^2 + (z_A - z_B)^2} \\ BC_{YZ} &= \sqrt{(y_B - y_C)^2 + (z_B - z_C)^2} \\ CA_{YZ} &= \sqrt{(y_C - y_A)^2 + (z_C - z_A)^2} \end{aligned} \tag{4.2}$$



(a) Variation of θ and ϕ



(b) Variation of θ and ϕ

Figure 4.6: Orientation in Spherical Coordinates.

$$\begin{aligned} AB_{XY} &= \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \\ BC_{XY} &= \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2} \\ CA_{XY} &= \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2} \end{aligned} \tag{4.3}$$

$$\begin{aligned} AB_{ZX} &= \sqrt{(z_A - z_B)^2 + (x_A - x_B)^2} \\ BC_{ZX} &= \sqrt{(z_B - z_C)^2 + (x_B - x_C)^2} \\ CA_{ZX} &= \sqrt{(z_C - z_A)^2 + (x_C - x_A)^2} \end{aligned} \tag{4.4}$$

In spherical coordinates these equations become as in equations 4.5, 4.6 and 4.7 respectively.

$$\begin{aligned} AB_{YZ} &= \rho\sqrt{3}\sqrt{1 - \cos^2(\theta)\cos^2(\phi + \tfrac{\pi}{3})} \\ BC_{YZ} &= \rho\sqrt{3}\sqrt{1 - \cos^2(\theta)\cos^2(\phi)} \\ CA_{YZ} &= \rho\sqrt{3}\sqrt{1 - \cos^2(\theta)\cos^2(\phi - \tfrac{\pi}{3})} \end{aligned} \tag{4.5}$$

$$\begin{aligned} AB_{XY} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\phi + \tfrac{\pi}{3})} = \rho\sqrt{3}\cos^2(\phi + \tfrac{\pi}{3}) \\ BC_{XY} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\phi)} = \rho\sqrt{3}\cos^2(\phi) \\ CA_{XY} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\phi - \tfrac{\pi}{3})} = \rho\sqrt{3}\cos^2(\phi - \tfrac{\pi}{3}) \end{aligned} \tag{4.6}$$

$$\begin{aligned} AB_{ZX} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\theta)\cos^2(\phi + \tfrac{\pi}{3})} \\ BC_{ZX} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\theta)\cos^2(\phi)} \\ CA_{ZX} &= \rho\sqrt{3}\sqrt{1 - \sin^2(\theta)\cos^2(\phi - \tfrac{\pi}{3})} \end{aligned} \tag{4.7}$$

Proof. looking at the first equation of AB from the set 4.2 where $AB_{YZ} = \sqrt{(y_A - y_B)^2 + (z_A - z_B)^2}$, we first replace the y and z coordinates of A and B with their values in spherical coordinates. This yields to,

$$AB_{YZ} = \sqrt{\big[\rho\sin(\phi)\sin(\theta) - \rho\sin(\phi+\tfrac{2\pi}{3})\sin(\theta)\big]^2 + \big[\rho\cos(\phi) - \rho\cos(\phi-\tfrac{2\pi}{3})\big]^2}$$
$$= \rho\sqrt{\sin^2(\theta)\big[\sin(\phi) - \sin(\phi+\tfrac{2\pi}{3})\big]^2 + \big[\cos(\phi) - \cos(\phi-\tfrac{2\pi}{3})\big]^2}$$

$$\textit{Let, } \begin{cases} \alpha - \beta = \phi \\ \alpha + \beta = \phi + \frac{2\pi}{3} \end{cases} \quad \textit{then, } \begin{cases} \alpha = \phi + \frac{\pi}{3} \\ \beta = \frac{\pi}{3} \end{cases}$$

using the following trigonometry identities:
$\cos\alpha\sin\beta = \frac{1}{2}\big[\sin(\alpha+\beta) - \sin(\alpha-\beta)\big]$
$\sin\alpha\sin\beta = \frac{1}{2}\big[\cos(\alpha+\beta) - \cos(\alpha-\beta)\big]$

$$AB_{YZ} = \rho\sqrt{\sin^2(\theta)\big[2\cos(\phi+\tfrac{\pi}{3})\sin(\tfrac{\pi}{3})\big]^2 + \big[2\sin(\phi+\tfrac{\pi}{3})\sin(\tfrac{\pi}{3})\big]^2}$$
$$= \rho\sqrt{3\sin^2(\theta)\cos^2(\phi+\tfrac{\pi}{3}) + 3\sin^2(\phi+\tfrac{\pi}{3})}$$
$$= \rho\sqrt{3}\sqrt{(1-\cos^2(\theta))\cos^2(\phi+\tfrac{\pi}{3}) + \sin^2(\phi+\tfrac{\pi}{3})}$$
$$= \rho\sqrt{3}\sqrt{-\cos^2(\theta)\cos^2(\phi+\tfrac{\pi}{3}) + \cos^2(\phi+\tfrac{\pi}{3}) + \sin^2(\phi+\tfrac{\pi}{3})}$$

yield to,

$$\boxed{AB_{YZ} = \rho\sqrt{3}\sqrt{1-\cos^2(\theta)\cos^2(\phi+\tfrac{\pi}{3})}}$$

The same reasoning can be done to derive the equations of the projections of BC_{YZ} and CA_{YZ}

$$BC_{YZ} = \sqrt{(y_B - y_C)^2 + (z_B - z_C)^2}$$

replacing with spherical coordinates we get,

$$\begin{aligned} BC_{YZ} &= \sqrt{\left[\rho\sin(\phi + \tfrac{2\pi}{3})\sin(\theta) - \rho\sin(\phi - \tfrac{2\pi}{3})\sin(\theta)\right]^2 + \left[\rho\cos(\phi + \tfrac{2\pi}{3}) - \rho\cos(\phi - \tfrac{2\pi}{3})\right]^2} \\ &= \rho\sqrt{\sin^2(\theta)\left[\sin(\phi + \tfrac{2\pi}{3}) - \sin(\phi - \tfrac{2\pi}{3})\right]^2 + \left[\cos(\phi + \tfrac{2\pi}{3}) - \cos(\phi - \tfrac{2\pi}{3})\right]^2} \end{aligned}$$

$$\textit{Let, } \begin{cases} \alpha - \beta = \phi + \frac{2\pi}{3} \\ \alpha + \beta = \phi - \frac{2\pi}{3} \end{cases} \quad \textit{then, } \begin{cases} \alpha = \phi \\ \beta = \frac{2\pi}{3} \end{cases}$$

using the following trigonometry identities:

$\cos\alpha\sin\beta = \frac{1}{2}\left[\sin(\alpha + \beta) - \sin(\alpha - \beta)\right]$

$\sin\alpha\sin\beta = \frac{1}{2}\left[\cos(\alpha + \beta) - \cos(\alpha - \beta)\right]$

$$\begin{aligned} BC_{YZ} &= \rho\sqrt{\sin^2(\theta)\left[2\cos(\phi)\sin(\tfrac{2\pi}{3})\right]^2 + \left[2\sin(\phi)\sin(\tfrac{2\pi}{3})\right]^2} \\ &= \rho\sqrt{3\sin^2(\theta)\cos^2(\phi) + 3\sin^2(\phi)} \\ &= \rho\sqrt{3}\sqrt{(1 - \cos^2(\theta))\cos^2(\phi) + \sin^2(\phi)} \\ &= \rho\sqrt{3}\sqrt{-\cos^2(\theta)\cos^2(\phi) + \cos^2(\phi) + \sin^2(\phi)} \end{aligned}$$

yield to,

$$\boxed{BC_{YZ} = \rho\sqrt{3}\sqrt{1 - \cos^2(\theta)\cos^2(\phi)}}$$

$$CA_{YZ} = \sqrt{(y_B - y_C)^2 + (z_B - z_C)^2}$$

replacing with spherical coordinates we get,

$$\begin{aligned} CA_{YZ} &= \sqrt{\left[\rho\sin(\phi)\sin(\theta) - \rho\sin(\phi - \tfrac{2\pi}{3})\sin(\theta)\right]^2 + \left[\rho\cos(\phi) - \rho\cos(\phi - \tfrac{2\pi}{3})\right]^2} \\ &= \rho\sqrt{\sin^2(\theta)\left[\sin(\phi) - \sin(\phi - \tfrac{2\pi}{3})\right]^2 + \left[\cos(\phi) - \cos(\phi - \tfrac{2\pi}{3})\right]^2} \end{aligned}$$

$$\textit{Let, } \begin{cases} \alpha - \beta = \phi \\ \alpha + \beta = \phi - \frac{2\pi}{3} \end{cases} \qquad \textit{then, } \begin{cases} \alpha = \phi - \frac{\pi}{3} \\ \beta = \frac{\pi}{3} \end{cases}$$

using the following trigonometry identities:

$\cos\alpha\sin\beta = \frac{1}{2}\left[\sin(\alpha+\beta) - \sin(\alpha-\beta)\right]$

$\sin\alpha\sin\beta = \frac{1}{2}\left[\cos(\alpha+\beta) - \cos(\alpha-\beta)\right]$

$$\begin{aligned} CA_{YZ} &= \rho\sqrt{\sin^2(\theta)\left[2\cos(\phi - \tfrac{\pi}{3})\sin(\tfrac{\pi}{3})\right]^2 + \left[2\sin(\phi - \tfrac{\pi}{3})\sin(\tfrac{\pi}{3})\right]^2} \\ &= \rho\sqrt{3\sin^2(\theta)\cos^2(\phi - \tfrac{\pi}{3}) + 3\sin^2(\phi - \tfrac{\pi}{3})} \\ &= \rho\sqrt{3}\sqrt{(1 - \cos^2(\theta))\cos^2(\phi - \tfrac{\pi}{3}) + \sin^2(\phi - \tfrac{\pi}{3})} \\ &= \rho\sqrt{3}\sqrt{-\cos^2(\theta)\cos^2(\phi - \tfrac{\pi}{3}) + \cos^2(\phi - \tfrac{\pi}{3}) + \sin^2(\phi - \tfrac{\pi}{3})} \end{aligned}$$

yield to,

$$\boxed{CA_{YZ} = \rho\sqrt{3}\sqrt{1 - \cos^2(\theta)\cos^2(\phi - \tfrac{\pi}{3})}}$$

We can apply the same reasoning and analysis on the second set (eq. 4.6) and third set (eq. 4.7) of equations to determine the projected distances onto the planes (XY) and (ZX).

□

Now that we have a simplified form of the projected distance, we can proceed by analyzing these forms in order to extract relationships between these projections and the shape of the geometry.

3.3 Extracting the Embodied Sphere Radius

Since the vertices of the marker are embedded in a spherical shape, our goal here is to find ρ because ρ can tell us something about the size of the sphere and thus we can extract the size of the marker. But what we wish to have and what is most important is to have an equation of ρ, which is function of the projected distances and is independent of the orientation. Because in reality the projection is captured on one plane only, we have to choose between one of these three sets of equation. We choose the plane (YZ) to be the plane parallel to the image plane. Therefore, we consider taking the first set (eq. 4.5) of these equations and try to find the value of ρ as function of these distances. What we noticed first is that these equations are not completely independent from each other. We tried here combining and developing these equations together and what we got is an astonishing result. We found, as our expectation, that ρ, the radius of the circumscribe sphere, can be written purely as function of the projected distance and is completely independent from the θ and ϕ which describe the orientation of the spherical marker. Eq 4.8 depicts the equation of ρ with its parameters a, b and c as function of the projected distance AB, BC and CA onto the plane (YZ). We follow the equation by the proof of its derivation.

$$\left.\begin{aligned}\rho &= \sqrt{\frac{-b-\sqrt{b^2-4ac}}{2a}}\end{aligned}\right\}\ \text{Virtual Sphere Model} \tag{4.8}$$

$$\begin{aligned}a &= -27\\ b &= 6\cdot(AB^2+BC^2+CA^2)\\ c &= (AB^2-BC^2-CA^2)^2-4\cdot CA^2\cdot BC^2\\ d &= 0\end{aligned}$$

Proof. From the simplified equation CA derived previously, we can develop its

form to write it as function of AB and BC and ρ as depicted below,

$$CA^2 = 3\rho^2\big[1 - \cos^2(\theta)\cos^2(\phi - \tfrac{\pi}{3})\big]$$

$$\begin{array}{c} \textit{however, } \cos(\phi - \frac{\pi}{3}) = 2\cos(\phi)\cos(\frac{\pi}{3}) - \cos(\phi + \frac{\pi}{3}) \\ \textit{and, } \cos^2(\phi - \frac{\pi}{3}) = \cos^2(\phi) + \cos^2(\phi + \frac{\pi}{3}) - 2\cos(\phi)\cos(\phi + \frac{\pi}{3}) \end{array}$$

This leads to,

$$\begin{aligned} \frac{AC^2}{3\rho^2} &= 1 - \cos^2(\theta)\big[\cos^2(\phi) + \cos^2(\phi + \tfrac{\pi}{3}) - 2\cos(\phi)\cos(\phi + \tfrac{\pi}{3})\big] \\ &= 1 - \cos^2(\theta)\cos^2(\phi) - \cos^2(\theta)\cos^2(\phi + \tfrac{\pi}{3}) - 2\cos^2(\theta)\cos(\phi)\cos(\phi + \tfrac{\pi}{3})\big] \\ &= 1 - \cos^2(\theta)\cos^2(\phi) - 1 + 1 - \cos^2(\theta)\cos^2(\phi + \tfrac{\pi}{3}) - 2\cos^2(\theta)\cos(\phi)\cos(\phi + \tfrac{\pi}{3})\big] \\ &= \frac{BC^2}{3\rho^2} - 1 + \frac{AB^2}{3\rho^2} - 2\sqrt{\cos^2(\theta)\cos^2(\phi)}\sqrt{\cos^2(\theta)\cos^2(\phi + \tfrac{\pi}{3})} \\ &= \frac{BC^2}{3\rho^2} - 1 + \frac{AB^2}{3\rho^2} - 2\sqrt{(-1)\left[-1 + \frac{BC^2}{3\rho^2}\right]}\sqrt{(-1)\left[-1 + \frac{AB^2}{3\rho^2}\right]} \end{aligned}$$

Re-arranging the equation gives us,

$$\begin{aligned} -3\rho^2 + BC^2 + AB^2 - AC^2 &= -2\sqrt{3\rho^2 - BC^2}\sqrt{3\rho^2 - AB^2} \\ -3\rho^2 + BC^2 + AB^2 - AC^2 &= -2\sqrt{9\rho^4 - 3\rho^2BC^2 - 3\rho^2AB^2 + BC^2AB^2} \end{aligned}$$

squaring both sides, we get

$$\big[-3\rho^2 + BC^2 + AB^2 - AC^2\big]^2 = 4\big[9\rho^4 - 3\rho^2BC^2 - 3\rho^2AB^2 + BC^2AB^2\big]^2$$

Developing and re-arranging the equation leads to,

$$\begin{array}{c} 9\rho^4 + \big[BC^2 + AB^2 - AC^2\big] - 6\rho^2\big[BC^2 + AB^2 - AC^2\big] \\ = \\ 4\big[9\rho^4 - 3\rho^2BC^2 - 3\rho^2AB^2 + BC^2AB^2\big] \end{array}$$

which gives,

$$-27\rho^4 + \big[6AC^2 - 6BC^2 - 6AB^2\big]\rho^2 + \big[AB^2 + BC^2 - AC^2\big]^2 - 4BC^2CA^2 = 0$$

and is of the form:

$$ax^2 + bx + c = d$$

if we let,

$$\begin{aligned} a &= -27 \\ b &= 6 \cdot (AB^2 + BC^2 + CA^2) \\ c &= (AB^2 - BC^2 - CA^2)^2 - 4 \cdot CA^2 \cdot BC^2 \\ d &= 0 \end{aligned}$$

we get a polynomial quadratic equation of ρ as function of the projected distances AB, BC and CA. And if we let again $X = \rho^2$, we get a second degree equation of the form,

$$-27X^2 + \left[6AC^2 - 6BC^2 - 6AB^2\right]X + \left[AB^2 + BC^2 - AC^2\right]^2 - 4BC^2CA^2 = 0$$

and has a solution of the form,

$$X = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Since $\rho = X^2$, this yield to the solution of ρ,

$$\boxed{\rho = \sqrt{\frac{-b - \sqrt{b^2 - 4ac}}{2a}}}$$

□

So far we have proven from the above demonstration that, for an equilateral triangle in 3D space, where the geometric location of its vertices lies on a sphere, we can derive a function that relates all the projected distances on one plane to a constant value, which represents the radius ρ of the embodied sphere. Furthermore, we validate carefully all our developed equations using Matlab. All the above results show perfect match with all the simulations and developed algorithms on Matlab. We plot below in figure 4.7 the value of ρ which we extract from eq. 4.8, using only the three projected distances AB, BC and CA for different orientation of the marker. The results show perfect match of ρ with the initial radius of the sphere.

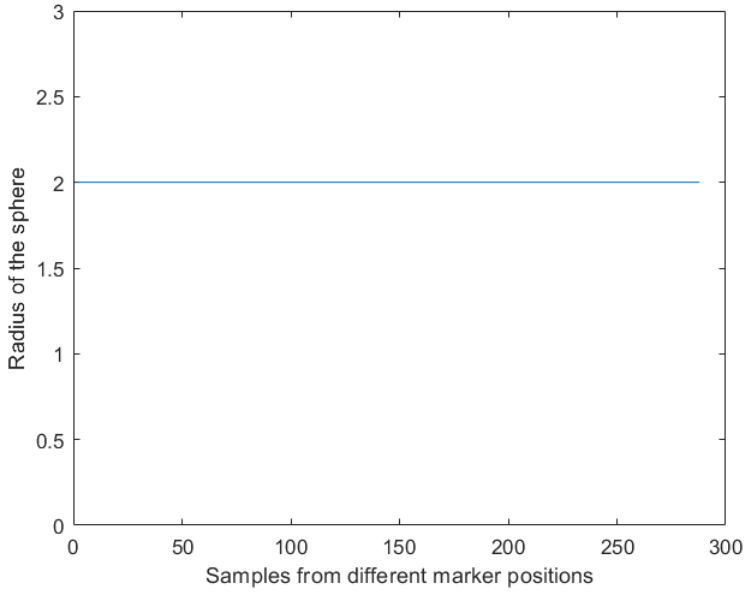


Figure 4.7: Constant value of the ρ extracted from the projected distances as we rotate the marker in all direction.

The polynomial equation of ρ, we derived here, will be integrated into our mathematical model to capture the size the of sphere using only the projected distances into the image plane.

Later on, in this chapter we will show as well, that the same concept applies to any triangular shape, where we demonstrate and generate the general form that captures the radius of the Virtual Circumscribed Sphere fitted into any three points in space, forming any triangular shape.

In the next section we apply foreshortening and distance correction to recover for the distortion from these effects. We then try to estimate the position of the triangular geometry. The results we achieved are demonstrated with simulations.

3.4 Recovery from Foreshortening Effect

In this section, we introduce the reader to our novel concept that we named the Virtual Sphere (VS). We use a triangular marker geometry to explore our first

position extraction method. This method lies within a concept of virtual sphere into which our marker virtually exists. Furthermore, we can say that the solution we will provide are fully deterministic. We develop and use a pure mathematical model based on perspective projection theorems and on postulates and axioms from fundaments of geometry published in the EUCLID'S ELEMENTS books collection (58).

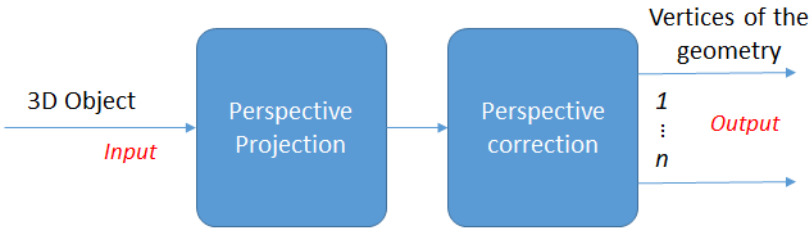


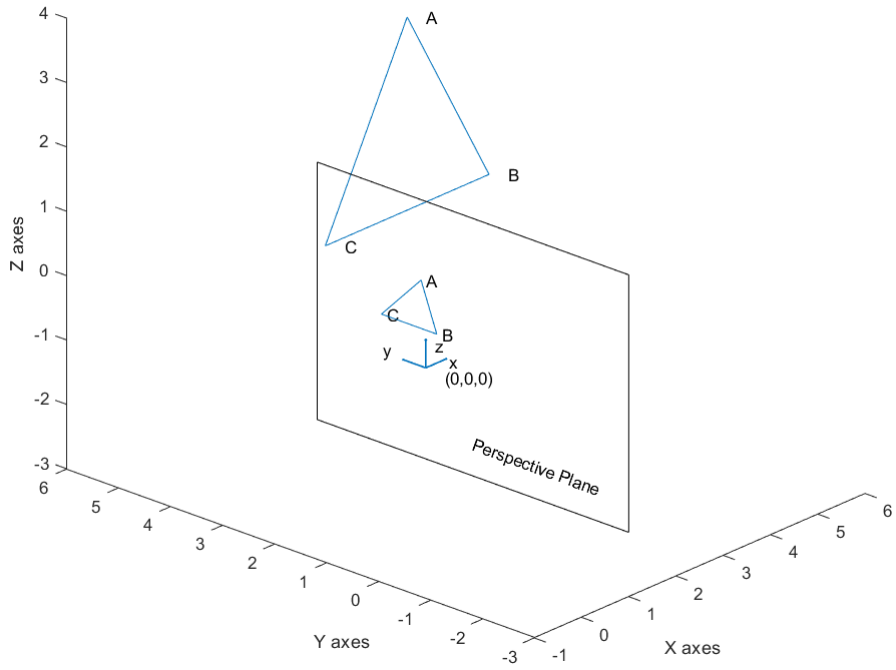
Figure 4.8: Workflow

We begin by placing our triangular marker in a 3D space and capture the first instance by recording the 2D correspondences of the vertices of the marker in the image plane as in figure 4.9 - (a). We then apply our perspective correction algorithm. Perspective correction is the transition phase between the perspective projection and parallel projection, since this correction process tends to restore the original form of the projection. Thus, we can consider our projections to be parallel after applying the corrections. One instance is enough to capture the position of the marker, as we will show later, but for the purpose of proving the concept we perform the simulation on different angles of the projection, as shown in figure 4.9. The sequence of the workflow is depicted in figure 4.8.

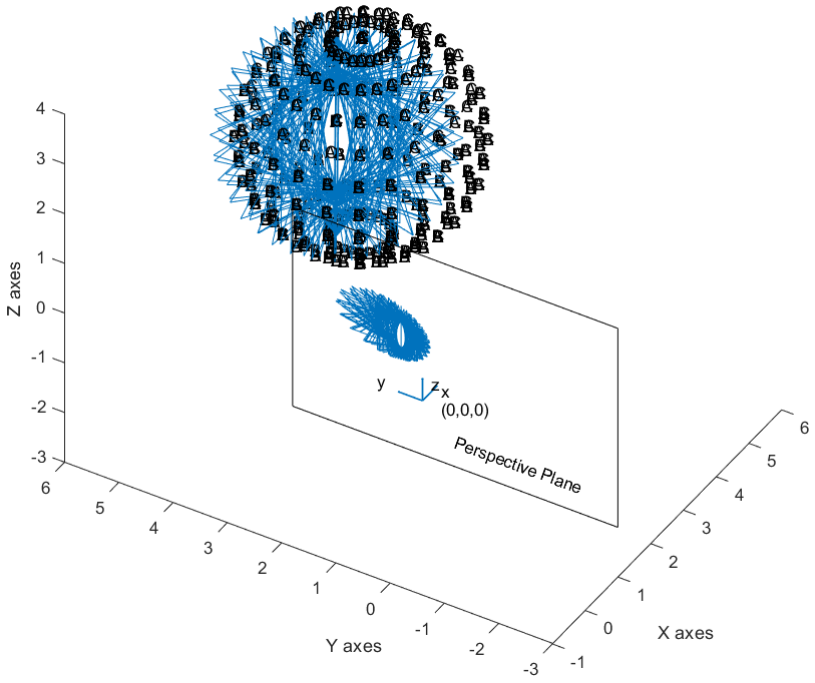
To prove the efficiency of the correction methods we developed, we run different simulations of perspective correction with different approaches:

(1) **Center-Of-Gravity projective rotation**
The COG approach, we locate the COG of the projected triangle and perform a projective rotation of the vertices with respect to the same point, the COG.

(2) **Mid-Arc projective rotation**
In this approach, we perform a mid-arc projective rotation for every segment separately.



(a) One instance of projection of the triangle into the image plane.

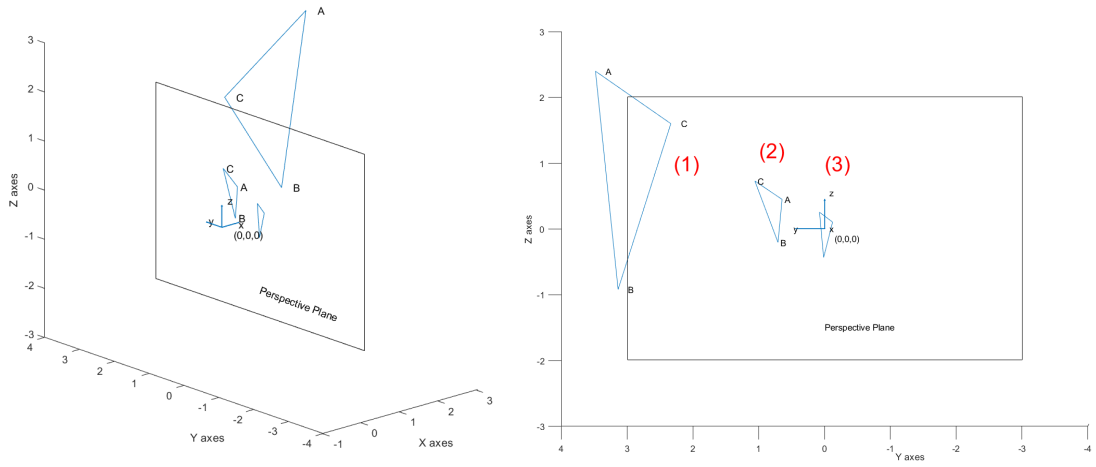


(b) Instances of perspective projection of the triangular marker.

Figure 4.9: Simulation of the triangle projection into the image plane.

(3) **Mid-Point of the triangle projective rotation**
In this approach we perform a projective rotation in order to have the middle vertices of the triangle coincident with the origin.

Figure 4.10 shows the results from the first approach. In figure 4.10 (a), we show the projection from perspective view side, while in figure 4.10 (b) we plot the view from the image plane (YZ) side. We mark by (1) the projection parallel of the original triangle, while triangle (2) depicts the perspective projection of the same triangle. Triangle (3) is the result of the projective rotation correction. We notice clearly that (1) and (3) are very similar except that A and C are swapped and this comes from the original perspective projection. But this is not an issue since what we care about is the shape and not the vertices labeling.



(a) Perspective view of the projection (b) Image plane view of the projection

Figure 4.10: Perspective projection and correction

To measure how accurate the correction is, we first feed all the three projected corrected triangles into the mathematical model to extract the radius of the three Virtual Spheres. What we expect to see is a constant variation of the radius as we rotate the rectangle in all direction.

We plot the result of the three radius extracted in figure 4.11. Since we did not add any correction for the distance effect yet, the mean for each set of data is different. But what is important to us is value of the mean square error or MSE.

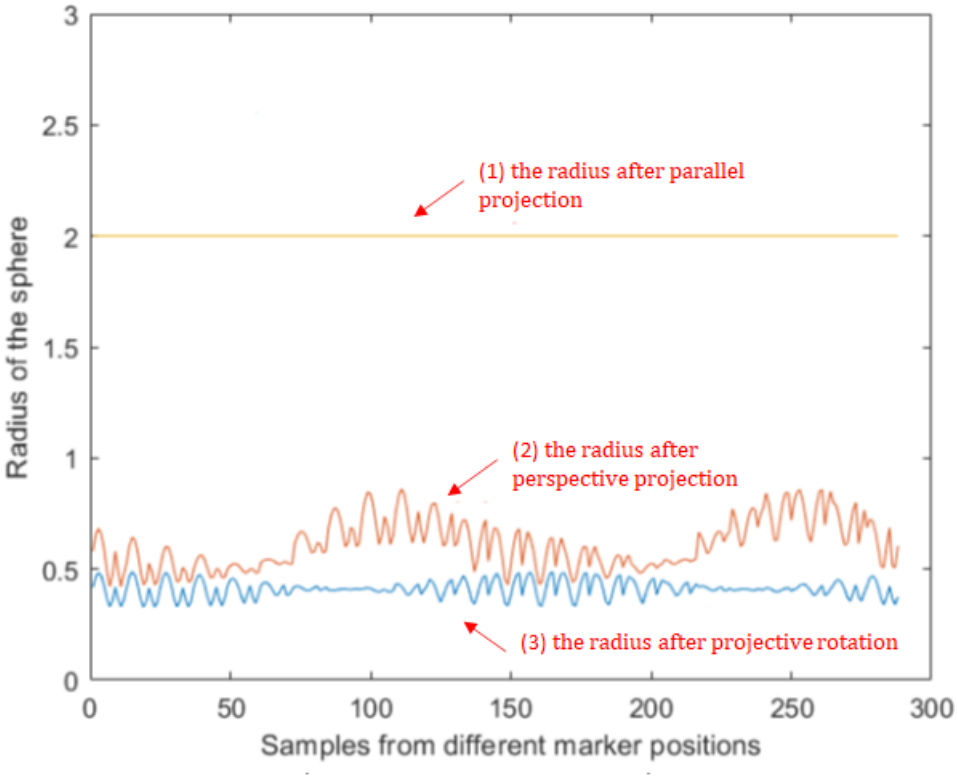


Figure 4.11: Virtual Sphere Radius

It is clear from the figure, that the MSE after correction in (3) is much smaller in plot (2) and it looks more like the plot of (1). Furthermore, we will show that these small fluctuations around the mean are due to the fact that there is no compensation for the pose effect.

Recovery from distance effect is simply made by multiplying all parameters by the ratios of the real radius of the VS over the extracted radius as shown in eq. 4.9.

$$\mu = \frac{\rho_{realradius}}{\rho_{extractedradius}} \tag{4.9}$$

We will proceed in the next section by extracting and analyzing the position of the marker in 3D space from the projected distances.

4 Position Estimation of the Marker in 3D

In the previous section we extracted all the necessary parameters to calculate the position of the 3D marker in 3D space.

We use the below workflow to extract the position of the Marker as shown in figure 4.12.

We develop a simple algorithm to extract the pose out of the Virtual Sphere (VS) model. The traditional Pinhole Pose Estimation Model is described in the earlier chapter. The difference between our model and the traditional model is that the later considers every point in space as an individual entity. We rather reckon points as belonging to the geometry and not as part of the real world. This makes the use of our marker strictly independent from its position in the real world. Our center is the geometry center of gravity (COG) and not any external reference point in the world. The link to the real world comes after and is optional.

We describe here the pose estimation model. The basic component is the Camera Pinhole Model, but instead of using directly the result out of the projections from

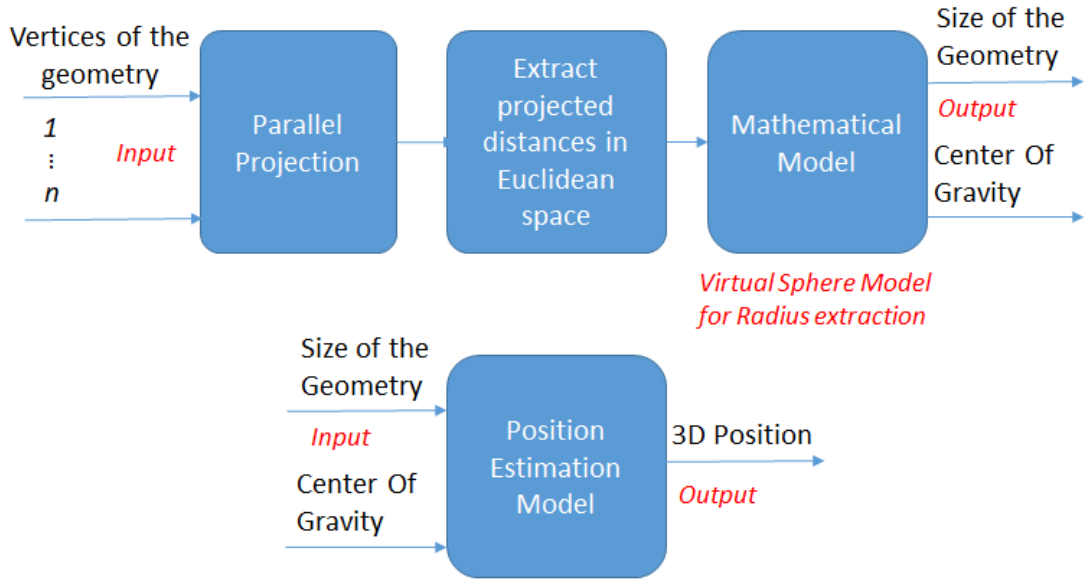


Figure 4.12: Workflow - Position Estimation

a Pinhole model, we add an intermediate correction model described previously and make use of these corrections to estimate the marker pose.

Now that we have the radius of the virtual sphere, which after correction is centered with respect to the view axis, we can project this radius perspectively into the image plane and calculate its distance as show in figure 4.13. This distance represents the real distance of the COG of the triangular marker with respect to the COP. We then use the initial projected triangle COG to define the directions where this marker lies as shown in figure 4.14. Combining the distance and the direction we can position our marker in 3D space.

From figure 4.13, the distance d to the center of the VS can be calculated as the ratios $d = \mu \cdot F$. μ is the ratio of the real radius over the projected radius. To calculate the coordinated of the center in the real world we simply multiply the coordinate of center of the initial projections by the factor $\frac{d}{d'}$. Here, d' represents the distance from COP to the COG of the projected triangle and is equal to $\sqrt{Gx^2 + Gy^2 + Gz^2}$ where G represents the COG of the projected triangle. We

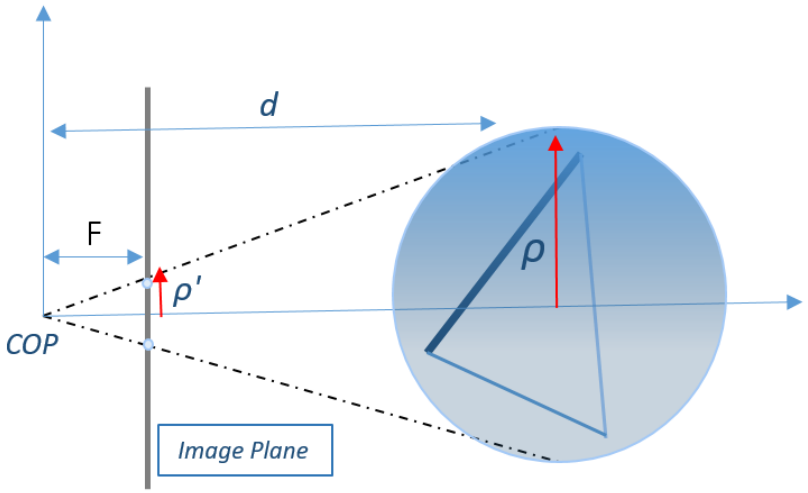


Figure 4.13: Distance From the VS

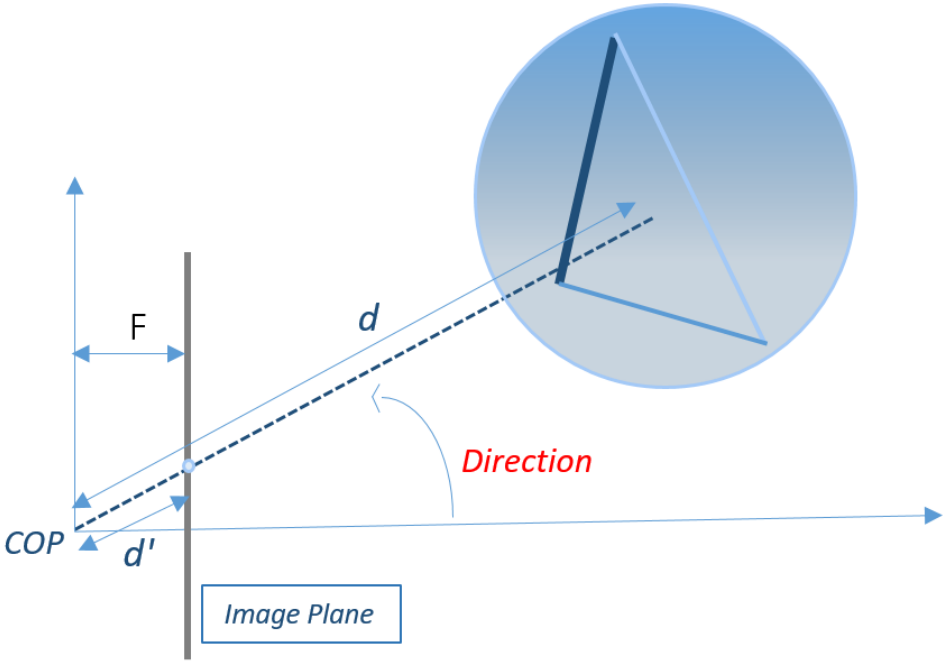


Figure 4.14: Direction of the VS

finally get the position equation as in eq 4.10.

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} \frac{\rho \cdot F}{\rho' \cdot d'} & 0 & 0 \\ 0 & \frac{\rho \cdot F}{\rho' \cdot d'} & 0 \\ 0 & 0 & \frac{\rho \cdot F}{\rho' \cdot d'} \end{bmatrix} \cdot \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} \tag{4.10}$$

We finally demonstrated with proof the derivation of the equations of the position estimation for the triangular marker. We run a simulation here for further proof of the equation before experimenting in real life.

We execute the following simulation:

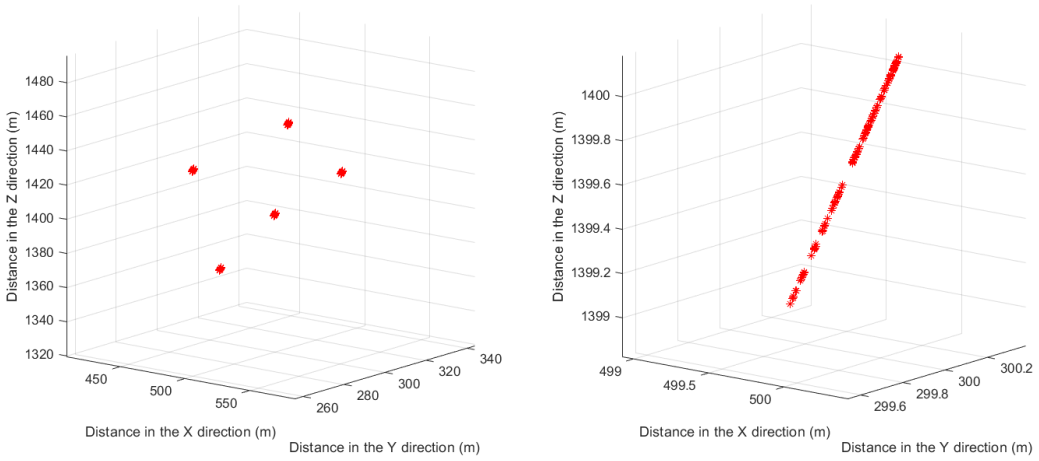
(1) We place a triangular marker circumscribed in a sphere of 4cm diameter in different positions anywhere in space within the field of view.

(2) For every position we capture 300 instances of the marker with different orientations for every instance.

(3) We then extract the position for all the instances and in all the different positions using our new model.

The results are shown in figure 4.15

Figure 4.15 (a) shows the plot of the positions of the Marker. We can see that for a particular position, even if we rotate the marker in all directions, the cloud point is very small. Figure 4.15 (b) shows the distribution of the cloud point of the marker placed at the position $[1400, 500, 300]$. These points are distributed linearly, since the radius of the marker is projected linearly. The same pattern is seen in all the 5 positions. This small margin error is due to the fluctuations described in the previous section in the value of ρ (the extracted radius). The error as we can see, is within a range below centimeters.



(a) Position Estimation of Marker from 5 different positions, and 300 different orientations.

(b) The generated position of the marker after orientation in 300 different poses.

Figure 4.15: Accurate Position extracted of the Marker.

5 Parallel Projection of any Triangular Shape

As promised previously, we extend the development to prove that the model explored in the previous section is also valid for any triangular shape, knowing its geometrical aspects and features.

In this section we explore the mathematical model that captures the radius of the circumscribed VS that passes through all the vertices of any form of triangle. What we intended to do is to find a function that relates the radius of the VS with the parallel projected distances between the vertices of the triangle, independently from its orientation. We used the same approach but this time we integrated the angle between the vertices into our initial equation. This makes the derivation much more complicated. We will call this model the Virtual Sphere 2 model (VS2) (16).

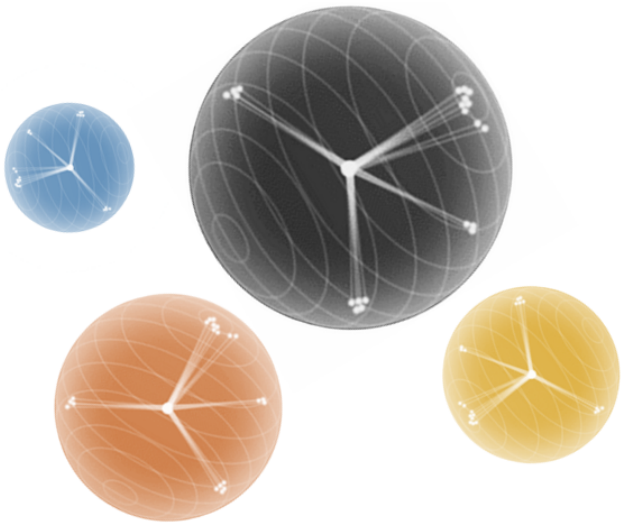


Figure 4.16: Virtual Spheres features

5.1 Representation in Spherical Coordinates

We consider here, that all three points are distributed on a spherical surface, on random positions.

As before, we use spherical coordinates to represent the position of vertices in space. If we let A, B and C to represent the vertices of the randomly generated triangle as in figure 4.18, and we denote by AB, BC and CA the real distances between its vertices, we can define these vertices in spherical coordinates (ρ, θ, ϕ) system with the convention described in figure 4.17 using the following coordinates equation 4.11:

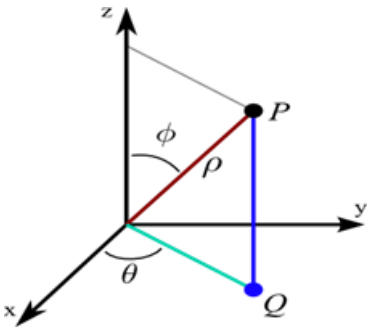


Figure 4.17: Spherical coordinate System.

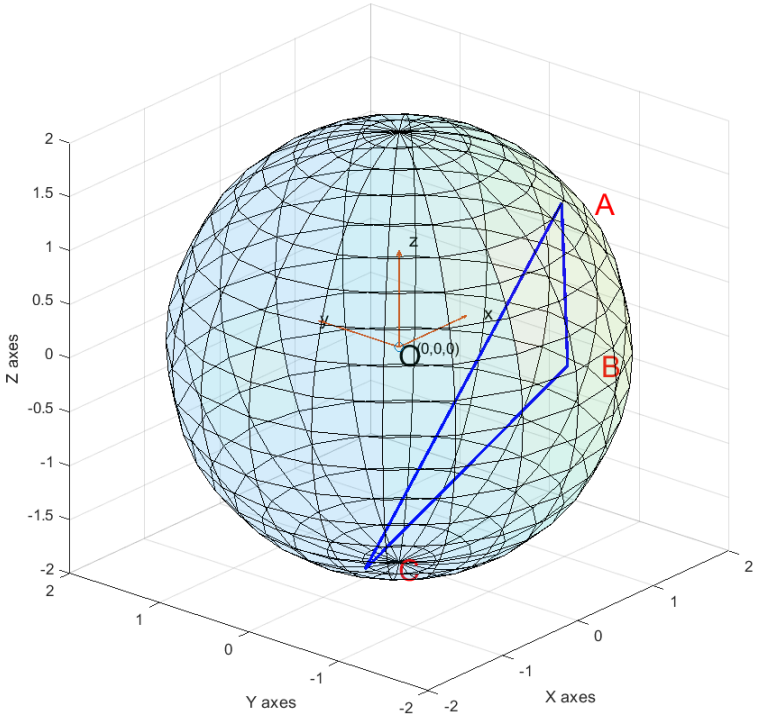


Figure 4.18: Equilateral Triangle in spherical coordinates.

$$
A = \begin{bmatrix} \rho\sin(\phi)\cos(\theta) + x_G \\ \rho\sin(\phi)\sin(\theta) + y_G \\ \rho\cos(\phi) + z_G \end{bmatrix}
$$

$$
B = \begin{bmatrix} \rho\sin(\phi+\alpha)\cos(\theta) + x_G \\ \rho\sin(\phi+\alpha)\sin(\theta) + y_G \\ \rho\cos(\phi+\alpha) + z_G \end{bmatrix} \qquad \begin{aligned} \rho &\in [0,\ \infty] \\ \theta &\in [0,\ 2\pi] \\ \phi &\in [0,\ \pi] \\ \alpha &\in [0,\ 2\pi] \\ \beta &\in [0,\ 2\pi] \end{aligned} \tag{4.11}
$$

$$
C = \begin{bmatrix} \rho\sin(\phi+\beta)\cos(\theta) + x_G \\ \rho\sin(\phi+\beta)\sin(\theta) + y_G \\ \rho\cos(\phi+\beta) + z_G \end{bmatrix}
$$

where ρ is radius of the sphere, θ and ϕ are the azimuthal and the polar angle

respectively, α and β are the angles formed by one point and the other two points, and x_G, y_G and z_G are the coordinates of its center. Defining our system in spherical coordinates allows us to orient the triangular marker in all the direction by varying θ and ϕ as in figure 4.19.

5.2 Parallel Projected Distances in Spherical Coordinates

From the coordinates equations of A, B and C we extract the parallel projected distances between all the three vertices in every fixed Euclidean plane using Euclidean distance. Equations 4.12, 4.13 and 4.14 represent the projected distances in the plane (YZ), (XY) and (ZX) respectively.

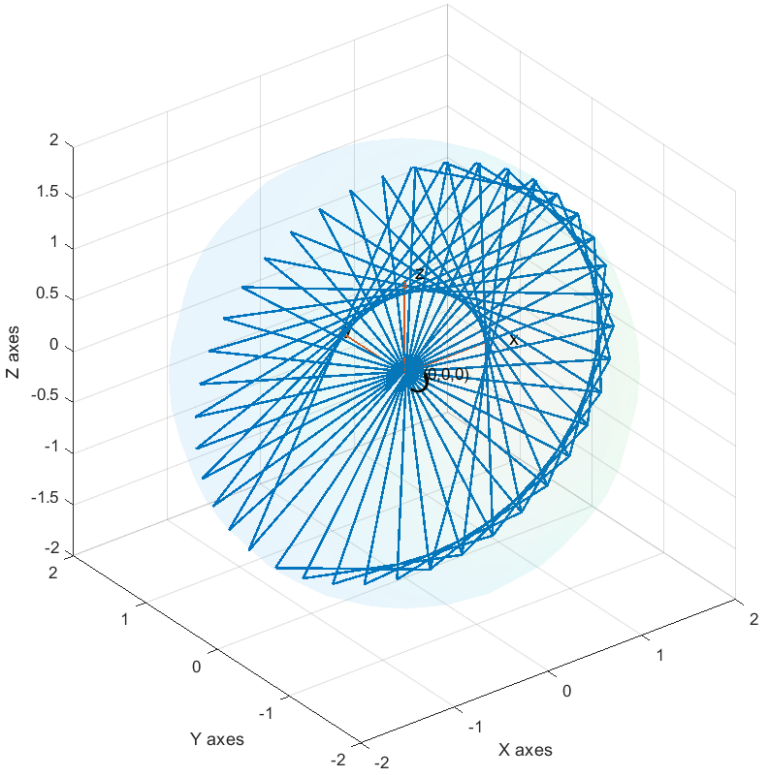
$$\begin{aligned} AB_{YZ} &= \sqrt{(y_A - y_B)^2 + (z_A - z_B)^2} \\ BC_{YZ} &= \sqrt{(y_B - y_C)^2 + (z_B - z_C)^2} \\ CA_{YZ} &= \sqrt{(y_C - y_A)^2 + (z_C - z_A)^2} \end{aligned} \tag{4.12}$$

$$\begin{aligned} AB_{XY} &= \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \\ BC_{XY} &= \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2} \\ CA_{XY} &= \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2} \end{aligned} \tag{4.13}$$

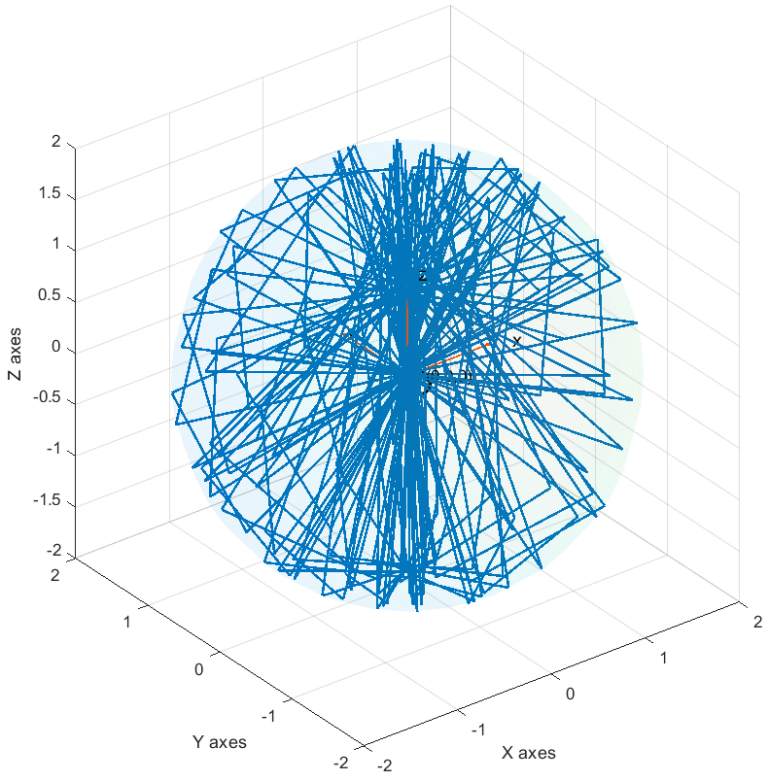
$$\begin{aligned} AB_{ZX} &= \sqrt{(z_A - z_B)^2 + (x_A - x_B)^2} \\ BC_{ZX} &= \sqrt{(z_B - z_C)^2 + (x_B - x_C)^2} \\ CA_{ZX} &= \sqrt{(z_C - z_A)^2 + (x_C - x_A)^2} \end{aligned} \tag{4.14}$$

In spherical coordinates these equations become as in equations 4.15, 4.16 and 4.17 respectively.

$$\begin{aligned} AB_{YZ} &= 2\rho \sin(-\tfrac{\alpha}{2}) \sqrt{1 - \cos^2(\theta)\cos^2(\phi + \tfrac{\alpha}{2})} \\ BC_{YZ} &= 2\rho \sin(\tfrac{\alpha-\beta}{2}) \sqrt{1 - \cos^2(\theta)\cos^2(\phi + \tfrac{\alpha+\beta}{2})} \\ CA_{YZ} &= 2\rho \sin(-\tfrac{\beta}{2}) \sqrt{1 - \cos^2(\theta)\cos^2(\phi + \tfrac{\beta}{2})} \end{aligned} \tag{4.15}$$



(a) Variation of θ and ϕ



(b) Variation of θ and ϕ

Figure 4.19: Orientation in Spherical Coordinates.

$$
\begin{aligned}
AB_{XY} &= 2\rho\sin(-\tfrac{\alpha}{2})\sqrt{1-\sin^2(\phi+\tfrac{\alpha}{2})} = 2\rho\sin(-\tfrac{\alpha}{2})\cos^2(\phi+\tfrac{\pi}{3}) \\
BC_{XY} &= 2\rho\sin(\tfrac{\alpha-\beta}{2})\sqrt{1-\sin^2(\phi+\tfrac{\alpha+\beta}{2})} = 2\rho\sin(\tfrac{\alpha-\beta}{2})\cos^2(\phi+\tfrac{\alpha+\beta}{2}) \\
CA_{XY} &= 2\rho\sin(-\tfrac{\beta}{2})\sqrt{1-\sin^2(\phi+\tfrac{\beta}{2})} = 2\rho\sin(-\tfrac{\beta}{2})\cos^2(\phi+\tfrac{\beta}{2})
\end{aligned} \tag{4.16}
$$

$$
\begin{aligned}
AB_{ZX} &= 2\rho\sin(-\tfrac{\alpha}{2})\sqrt{1-\sin^2(\theta)\cos^2(\phi+\tfrac{\alpha}{2})} \\
BC_{ZX} &= 2\rho\sin(\tfrac{\alpha-\beta}{2})\sqrt{1-\sin^2(\theta)\cos^2(\phi+\tfrac{\alpha+\beta}{2})} \\
CA_{ZX} &= 2\rho\sin(-\tfrac{\beta}{2})\sqrt{1-\sin^2(\theta)\cos^2(\phi+\tfrac{\beta}{2})}
\end{aligned} \tag{4.17}
$$

Proof. looking at the first equation of AB from the set 4.12 where $AB_{YZ} = \sqrt{(y_A-y_B)^2+(z_A-z_B)^2}$, we first replace the y and z coordinates of A and B with their values in spherical coordinates. We then proceed with the development of these equations. For further proof on the derivation of equations, the reader can follow the same methodology and apply the same reasoning as in section 3 for all the sets of equations. □

5.3 Extracting the Embodied Sphere Radius

Having a simplified form of the projected distance, we can proceed by analyzing these forms in order to extract relationships between these projections and the radius of the geometry.

In reality, the projection is captured on one plane only. Thus, we have to choose between one of these three sets of equations. We choose the plane (YZ) to be the plane parallel to the image plane, therefore, we consider taking the first set (eq. 4.15) of these equations and try to find the value of ρ as function of these distances.

An important notice here is that these equations are not completely independent from each other. We tried here combining and developing these equations together to find the relationship between ρ and the projected distances.

Just like before, what we found is that the radius of the circumscribed sphere can be written purely as function of the projected distance, and is completely independent from the θ and ϕ, which describe the orientation of the spherical marker. It is only dependent of the distances in addition to the new parameters, which represent the angles between the vertices α and β.

Equation 4.18 is the solution ρ of the second degree equation with its parameters a, b, c and d as function of the projected distance AB, BC, CA, α and β onto the plane (YZ). We follow the equation by the proof of its derivation.

$$
\left.
\begin{aligned}
\rho &= \sqrt{\frac{-b-\sqrt{b^2-4ac}}{2a}}
\end{aligned}
\right\} \text{Virtual Sphere 2 Model} \tag{4.18}
$$

$$
\begin{aligned}
a &= \left[(U-1)^2 - V^2 - W^2\right]^2 - 4\cdot V^2 \cdot W^2 \\
b &= -2(U-1)^4 \cdot \frac{BC^2}{w} - 2V^4 \cdot \frac{AB^2}{u} - 2W^4 \cdot \frac{CA^2}{v} \\
&\quad + 2(U-1)^2 \cdot V^2 \cdot \left[\frac{AB^2}{u} + \frac{BC^2}{w}\right] \\
&\quad + 2(U-1)^2 \cdot W^2 \cdot \left[\frac{BC^2}{w} + \frac{CA^2}{v}\right] \\
&\quad + 2V^2 \cdot W^2 \cdot \left[\frac{AB^2}{u} + \frac{CA^2}{v}\right] \\
c &= \left[(U-1)^2 \cdot \frac{BC^2}{w} - V^2 \cdot \frac{AB^2}{u} - W^2 \cdot \frac{CA^2}{v}\right]^2 \\
&\quad - 4V^2 \cdot W^2 \cdot \frac{AB^2}{u} \cdot \frac{CA^2}{v} \\
d &= 0 \\
U &= 4\cos^2(\tfrac{\beta}{2}) - 4\cos^2(\tfrac{\alpha}{2}) + 1 \\
V &= -4\cos(\tfrac{\beta}{2}) + 8\cos^2(\tfrac{\alpha}{2})\cos(\tfrac{\beta}{2}) - 4\cos(\tfrac{\alpha}{2})\cos(\tfrac{\alpha-\beta}{2}) \\
W &= -2\cos(\tfrac{\alpha}{2}+\beta) + 2\cos(\frac{\alpha}{2}) \\
u &= 4\sin^2(-\tfrac{\alpha}{2}) \\
v &= 4\sin^2(-\tfrac{\beta}{2}) \\
w &= 4\sin^2(\tfrac{\alpha-\beta}{2})
\end{aligned}
$$

Proof. Our goal here is to write one of the distance equation as function of the others, therefore, since BC contains both α and β, what we wish to do then is to write $\cos(\phi + \frac{\alpha+\beta}{2})$ as function of both $\cos(\phi + \frac{\alpha}{2})$ and $\cos(\phi + \frac{\beta}{2})$ as follow:

$$\cos(\phi + \tfrac{\alpha+\beta}{2}) = U\cos(\phi + \tfrac{\alpha+\beta}{2}) + V\cos(\phi + \tfrac{\alpha}{2}) + W\cos(\tfrac{\beta}{2})$$

We found out after a lot of development using the trigonometry identities that this assumption is true and that U, V and W can be obtained.

$$\begin{cases} U = 4\cos^2(\frac{\beta}{2}) - 4\cos^2(\frac{\alpha}{2}) + 1 \\ V = -4\cos(\frac{\beta}{2}) + 8\cos^2(\frac{\alpha}{2})\cos(\frac{\beta}{2}) - 4\cos(\frac{\alpha}{2})\cos(\frac{\alpha-\beta}{2}) \\ W = -2\cos(\frac{\alpha}{2} + \beta) + 2\cos(\dfrac{\alpha}{2}) \end{cases}$$

we then square both sides and multiply by $\cos^2(\theta)$

$$\cos^2(\theta)\left[\quad\right]^2 = \cos^2(\theta)\left[\quad\right]^2$$

Having this form, we can replace AB, BC and CA with their appropriate forms in this equation as we did in the VS model section previously, to get our quadratic equation of ρ as function of the projected distances AB, BC and CA. And if we let again $X = \rho^2$, we get a second degree equation which has one of its solution of the form,

$$X = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Since $\rho = X^2$, this yield to the solution of ρ,

$$\boxed{\rho = \sqrt{\frac{-b - \sqrt{b^2 - 4ac}}{2a}}}$$

□

We would like to mention here that all these equation has been validated carefully on Matlab in order to be sure of our solution. All the above results show perfect

match with all the simulations and developed algorithms on Matlab. We plot below in figure 4.20 the value of ρ which we extract from eq. 4.18 using only the three projected distances AB, BC and CA and the angles α and β for different orientation of the marker. The results show perfect match of ρ with the initial radius of the sphere.

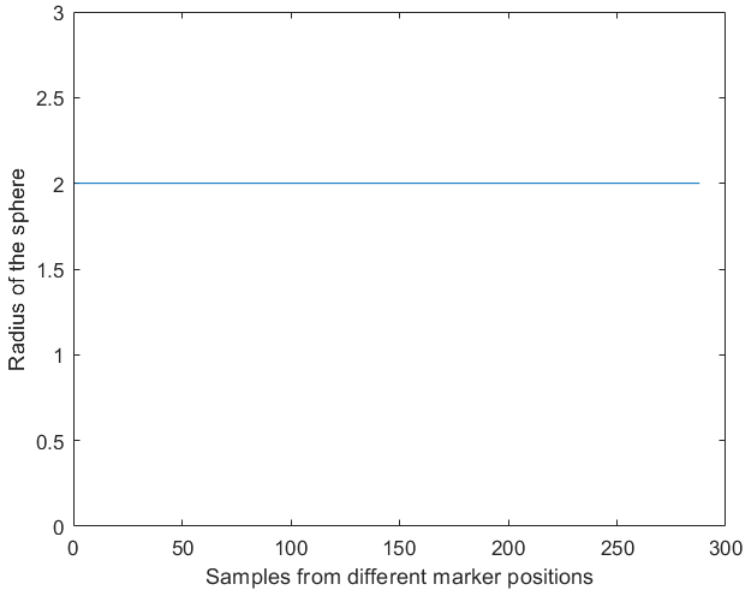


Figure 4.20: Constant value of the ρ extracted from the projected distances as we rotate the marker in all direction.

The polynomial equation of ρ, we derived here, will be integrated into our VS2 Model to capture the size the of sphere using only the projected distances into the image plane.

6 Summary and Discussion

In this Chapter we developed and demonstrated two models that capture sizes of Virtual Circumscribed Sphere fitted into equidistant points, VS model, and non equidistant points, VS2 model. These two purely mathematical models and they are perfect models that can be used when dealing with parallel projection setup.

We used these two models as part of the process in calculating the position of the center of the smallest embodied sphere on which the three vertices of a triangle are lying.

As a first step, in section 3 and 5 in this chapter, we showed that the radius of the real circumscribed sphere can be exactly extracted regardless of the position and orientation of the same triangle formed by the three vertices and regardless of the shape of the triangle, in parallel projection.

In the earlier chapter, we demonstrated the passage to reduce the error in image measurements because of perspective effects using the projective rotation. As a second step, in section 3.4 in this chapter, we apply the projective rotation formulas to eliminate the deterministic error coming from foreshortening perspective projection effect. Performing this correction allows us to calculate the real radius of the circumscribed sphere. We showed at the end of this section the derivation of the radius of the circumscribed sphere in three different cases:

(1) parallel projection,

(2) perspective projection,

(3) perspective projection followed by projection rotation correction.

We showed from the plot, of the extracted radius considering different orientation of the triangular target, that after performing a perspective projection followed by projective rotation, we tend eliminate the foreshortening error and the MSE from curve (3) becomes smaller and follows a symmetrical pattern. This symmetrical MSE comes from the pose effect of perspective projection.

Finally, in section 4, we demonstrated with proof of derivation of the equations and with simulations, the extraction the 3D position for the triangular target.

As a conclusion, we integrated the correction and the VS models in a novel method for relative position calculation which has given very robust and deter-

ministic results as we showed figure 4.15. The robustness is measured by the precision of the captured position as shown in figure 4.15 (a) and the accuracy of the measurements depicted in figure 4.15 (b). We also concluded that the precision factor is directly related to the size of the marker regardless of its position in space, while the accuracy range depends on the pose perspective effect factor discussed in chapter 3.

Furthermore, the model considered is a simple Pinhole model without additional distortions modeling. The only intrinsic parameter considered in this case is the focal distance F which adds no uncertainty to our model. Real experiments are done with complete real models that assimilate for the additional distortions as we will see in the later chapters.

Further investigations are presented in the next chapter, where we explore experimental work conducted throughout the work, for proving and validating our equation in a realistic environment. We build and use real markers on different benchmark models to test the behavior in real world with real cameras.

Chapter 5

Experimentations on Real Markers

Contents

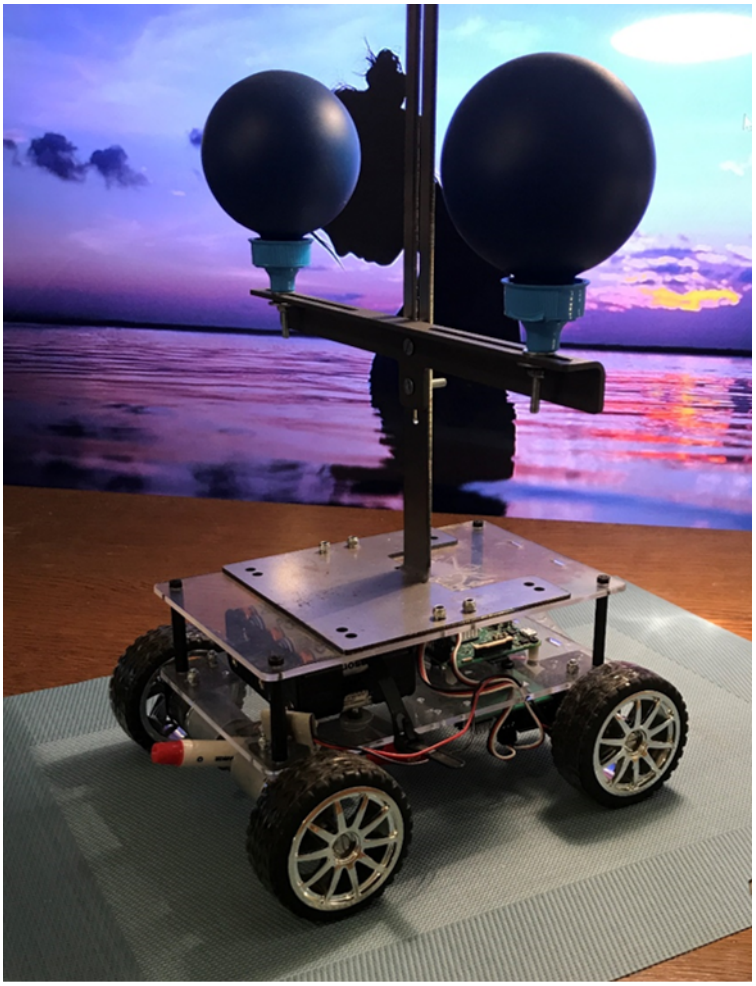


Figure 5.1: Perspective Marker Prototype.

1 Chapter Summary

Experimental results in this chapter serve as evidence to demonstrate the advantages of the proposed methods and concepts on the feasibility level, the implementation point of view and the potential use cases. We will perform scientific procedures and present experimental work from building the marker to constructing benchmarks related for different cases and methods.

In the next section we illustrate some of the materials and tools used and developed throughout the work.

2 Setting up the Experiments

We proceed in this section by first illustrating some of material used and prototypes developed. Figure 5.2 shows a set of designs of hardware, equipment and instruments used during experimentation.

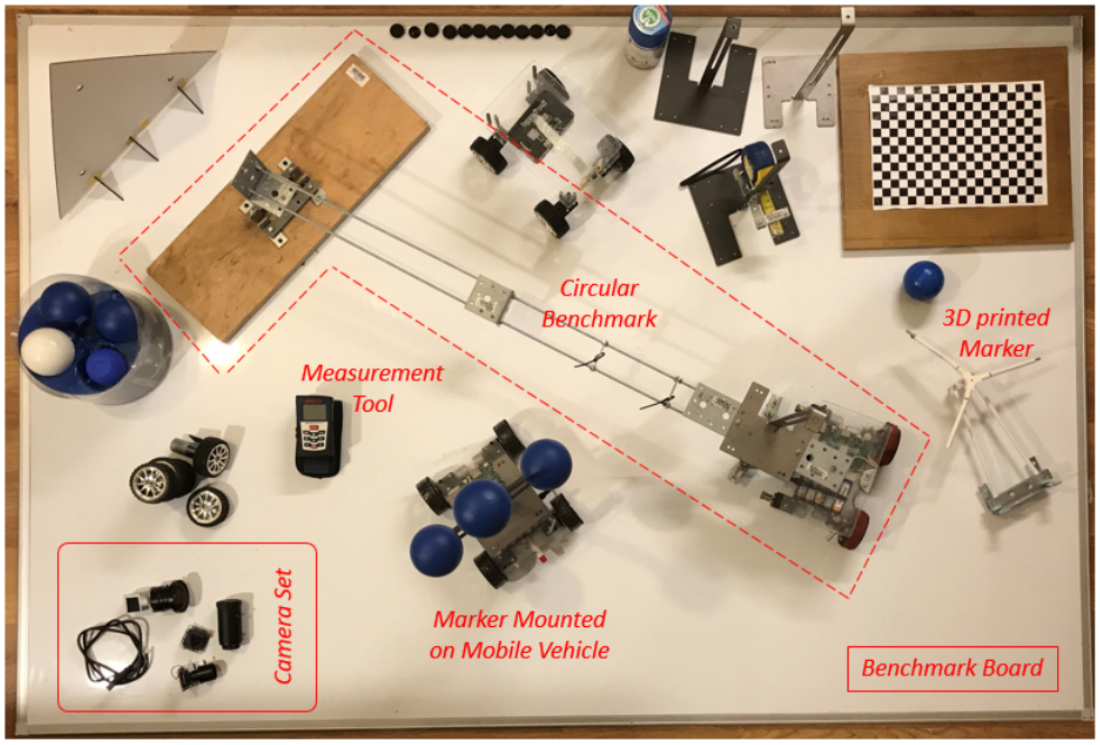


Figure 5.2: Designed hardware prototypes and instruments.

The hardware and software list can be grouped as follow:

(1) **Camera Set** - For experimenting in different environment and for different considerations like cost, range and accuracy, we assembled and gathered different camera sensors combined with different lenses. The list of cameras used is shown in table 5.1.

(2) **Benchmark Platforms** - Measuring and comparing the results require having true reference points, from which we can truly define the accuracy and the precision of the system. For this purpose we designed three concepts for measuring our data with respect to the ground truth. These concepts are il-

Camera Set			
Model	**Focal Len. (mm)**	**Pixel-Size (μm)**	**Other Specs.**
Picamera v2	3.04	1.12x1.12	• Resolution: 8MP • Focus Type: Fixed • Price < 25EUR
Basler Industrial acA2040	6.0	5.5x5.5	• Resolution: 4MP • Frame Rate: 90fps • Shutter: Global Shutter
Logitech HD C270	4.0	2.4x2.4	• Focus Type: Fixed • Field of View (FOV): 60° • Optical Resolution (True): 1280 x 960 1.2MP
CCTV Cameras (3x)	3.5	1.4x1.4	• Resolution: 3,4,8 MP • Optical Zoom: 2.8-12mm • IR shutter • Electronic rolling shutter / Frame exposure

Table 5.1: Cameras sensors specs.

lustrated more in detail in sections 2.1, 2.2 and 2.3 respectively, for the below listed benchmarks.

- The **Linear-Benchmark Platform** provides ground truth for measuring the distance of the marker with respect to the COP.
- The **Circular-Benchmark Platform** is used to benchmark the position when the marker is subject to rotation and circular motions.
- The third platform is the **Grid-Benchmark Platform** which we designed to show in real-time the accuracy and precision of the estimated position of the marker when it is freely moving within the FOV, referencing the ground truth. More details on the concept and benchmark operation process are depicted in section 2.3

In all the three platforms we place the camera sensor at known position from the center of the benchmark. We use the camera with its embedded software to capture images of the marker and extract its relative position.

(3) **Mobile Vehicle** - For navigating the marker around, we built three small vehicles. One that carries the camera sensor and the embedded PC, a second one that can carry the marker for free navigation and a third one that fits into the Circular Platform for turning the marker. Figure 5.2 illustrates the three vehicles.

(4) **Markers** - Throughout the development we built many markers that fit the purpose of the experimentation. We used laser cut markers, 3D printed markers and metal bent markers in different versions. We present some images of the used markers in section 2.5.

(5) **Tools and equipment** - We list here some tools and hardware used like:

Raspberry Pi as embedded PC

Bosch DLE 70 Professional range finder

Camera calibration board (14x20 squares)

Perforated metal shafts, PVCs, Spherical balls,

Infrared transmitters, extra camera lens

BNO055 IMU sensors, tinker board

Multi-Color Variable LED bulbs

(6) **Third-Party Softwares** - We list here some third-party softwares that were used for the developement:

- Matlab + Matlab toolboxes + VRML2
- ORBSLAM2 - Open Source SLAM
- openCV Libraries
- Solidworks
- Ubuntu16.04, Windows Machine, Raspbian Jessy

Further details on the benchmarks operation and architectures and on the markers with the mobile vehicles are presented in the next section.

2.1 Linear-Benchmark Platform

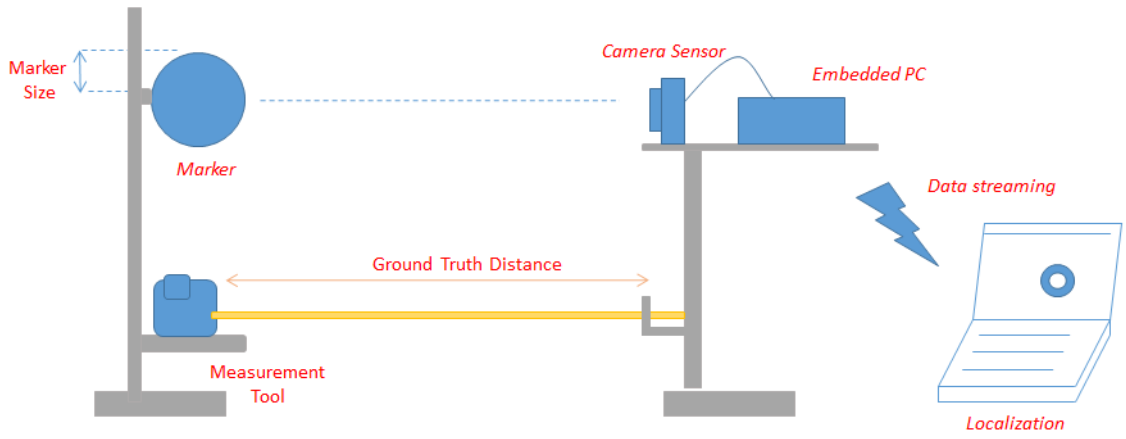


Figure 5.3: The concept of the Linear-Benchmark Platform

The linear benchmark we designed, consists of a simple architecture that allows us to measure the position relative to the distance of the marker aligned with focal axis of the camera sensor. We depict the concept in figure 5.3. The workflow diagram is also shown in figure 5.4. As shown in diagram, the system requires that we input the type of the geometry and camera distance to extract our ground truth. We use equation 2.8 to extract the depth d. Knowing the depth of the target, its position P can be derived considering that the target is centered. The position P becomes $P(0, 0, -d)$.

Figure 5.5 depicts the hardware of the platform where we see the camera sensor and the spherical marker aligned and are co-linear with the focal axis.

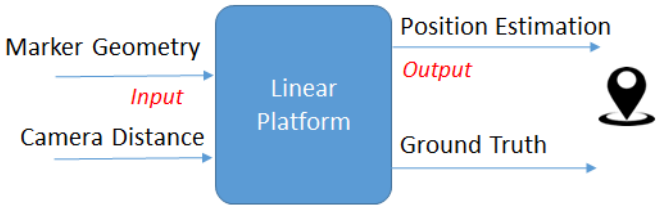


Figure 5.4: The workflow of the Linear-Benchmark Platform

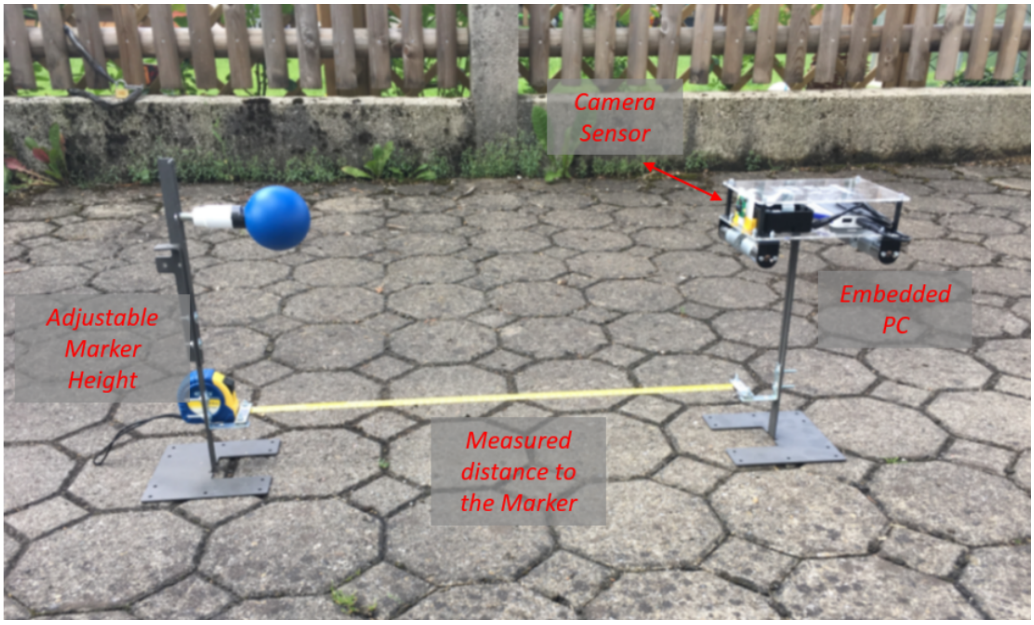


Figure 5.5: The hardware architecture of the Linear-Benchmark Platform

We designed this kind of architecture to help us in evaluating the distance effect on the marker projection as the marker moves straight away from the camera sensor and aligned with the focal axis. Furthermore, using this concept, we can identify and classify the marker used according to the following criteria:

- the marker geometry with respect to the application.
- the marker size with respect to the camera sensor resolution.
- the marker color for better detection and processing in the images.
- the accuracy of measurements with respect to distances.

2.2 Circular-Benchmark Platform

The circular platform is intended to measure how accurate are the circles formed by the data plots. This gives us a feedback about the accuracy of the system when the marker is oriented or in rotation motion. The workflow is depicted in figure 5.6

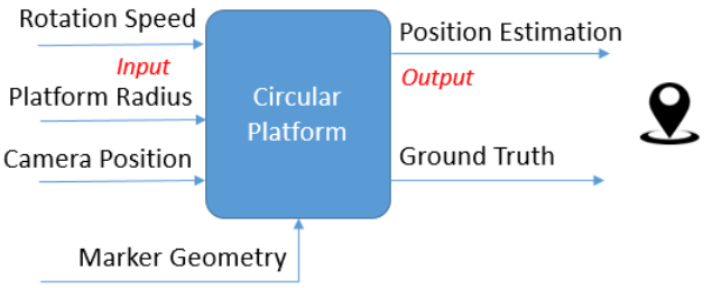


Figure 5.6: Workflow of the Circular-Benchmark Platform

Figure 5.7 shows the concept behind this benchmark. It consists of a Circular Platform on which the marker is attached and can rotate around a center point. Knowing the real radius of the platform, we can compare the radius of the extracted camera sensor data with the real one. This benchmark is designed to capture the pose effect on the marker's position for the purpose of testing the efficiency of our algorithms.

To generate the position of the target, we first extract the vertices of the markers the image. The vertices are located using the hough transform technique after applying image enhancement methods. We then apply the projective rotation on every side of the marker using equation 3.15 followed by the VS model equation 4.8 developed in chapter 4. Finaly, we apply equation 4.10 to calculate the position.

Furthermore, figure 5.8 shows real image of the benchmark hardware in operation. We show here the marker fixed on a mobile robot with constraint circular motion. The distance to the center can be adjusted to fit the need of the experiment. We fit the system with damping system in order to absorb the vibration coming from the uneven surface and in order to insure good grip with the floor. More pictures related to the benchmark shown in figure 5.9.

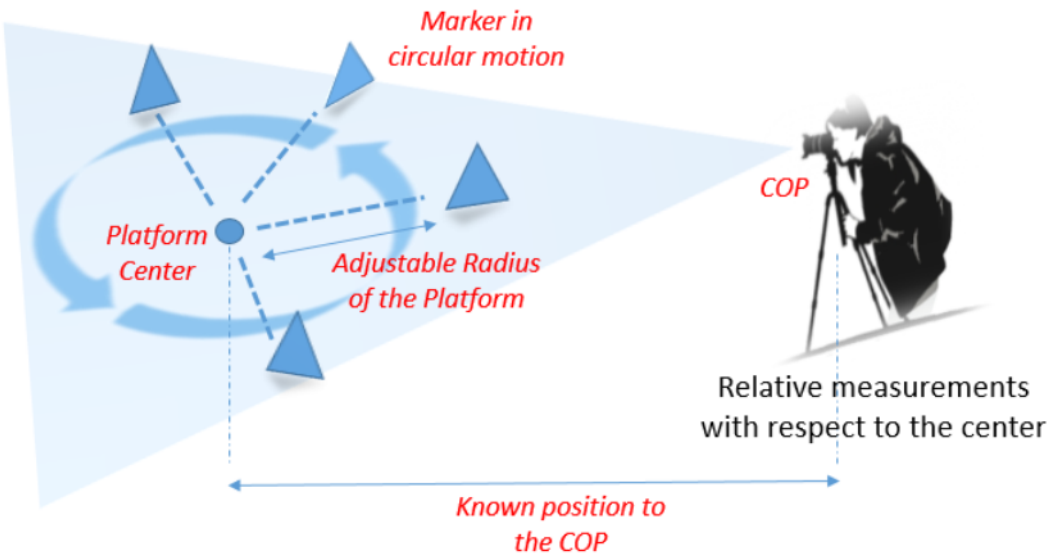


Figure 5.7: Concept of the Circular-Benchmark Platform

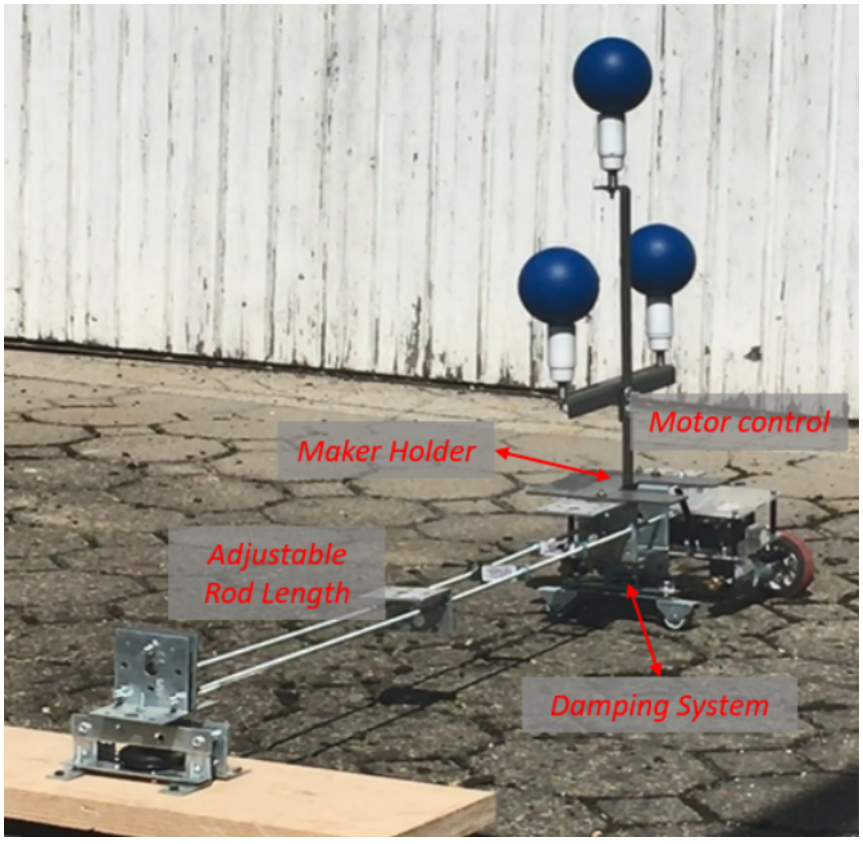


Figure 5.8: Hardware and design of the Circular-Benchmark Platform

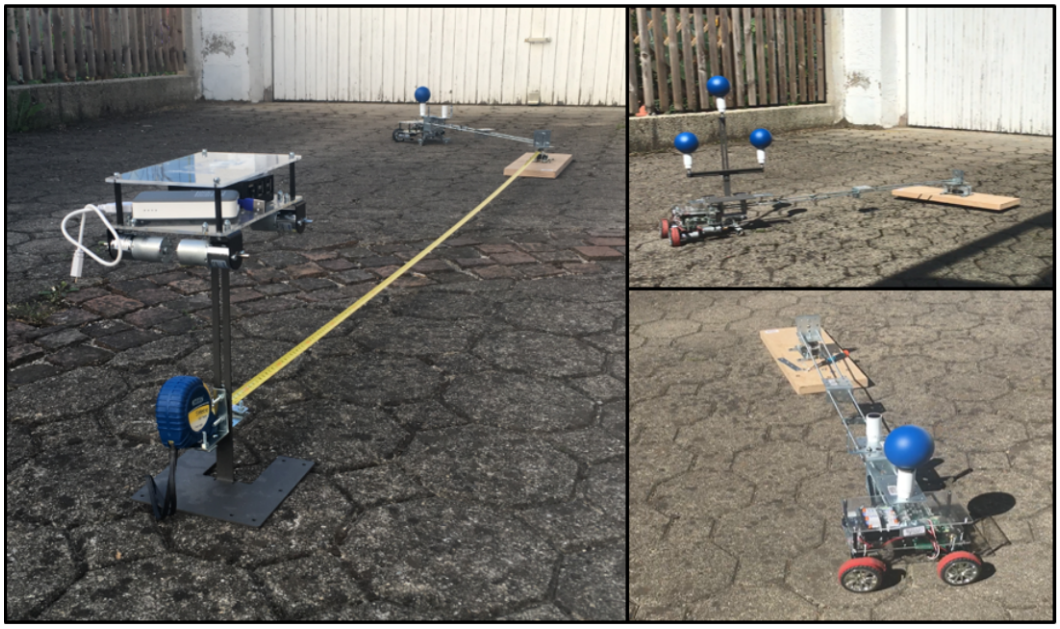


Figure 5.9: Circular Platform in operation with different markers.

2.3 Grid-Benchmark Platform

The benchmark present here is used for real-time measurement data where we put no constraints on the marker's motion. The goal is to capture the ground truth of the marker when navigating freely within the FOV of the camera sensor. The concept is illustrated in figure 5.10 and the workflow is presented in figure 5.11.

The system consists of two phases. The first phase is for capturing the data while the second one is for capturing the ground truth. We use a meshed white board (1.8mx1.2m) on which a mobile vehicle with a marker can mark its trajectory while navigating. We use the marked trajectory as evidence of the robot position while moving. The system works as follow:

We first feed the system with the initial position of the camera placed away from the meshed white board. Optionally, we can input the frame rate at which position is captured. Since the platform is suited to work with different markers geometry, we can specify the marker geometry as input to the system as well. As a first stage, the robot starts navigating while the camera sensor is capturing the

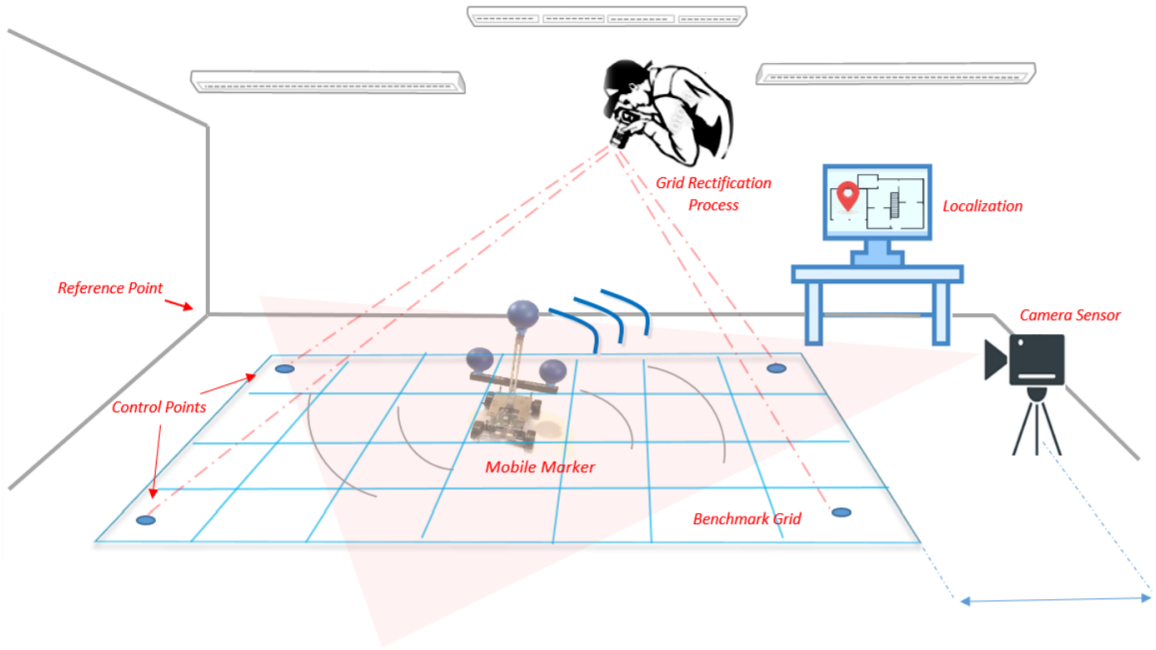


Figure 5.10: Concept of the Grid-Benchmark Platform.

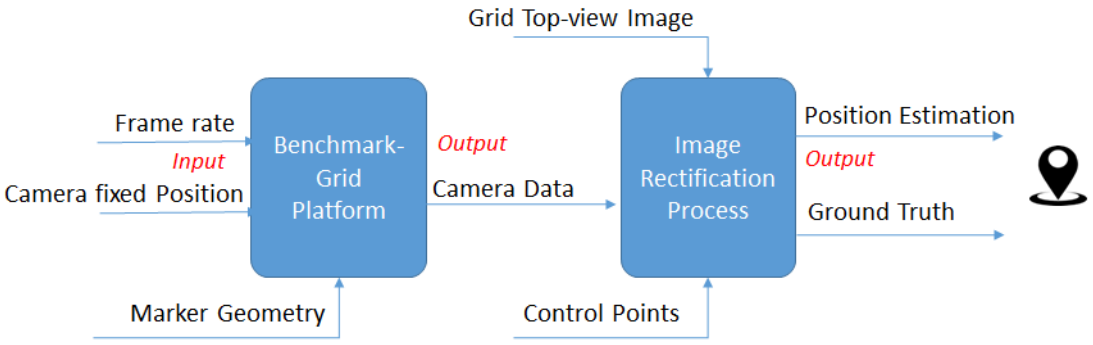


Figure 5.11: Workflow of the Grid-Benchmark Platform.

marker's position. Data from the camera are plotted on the GUI in real-time. Simultaneously, while navigating, the robot is marking its trajectory on the meshed board. To do so, we design a mechanical unit, fitted below the mobile robot, that marks its trajectory with a red marker while navigating on a board. We show in figure 5.12 a real-time position estimation. What is shown in this figure are the plot of the camera data and robot in motion marking its trajectory.

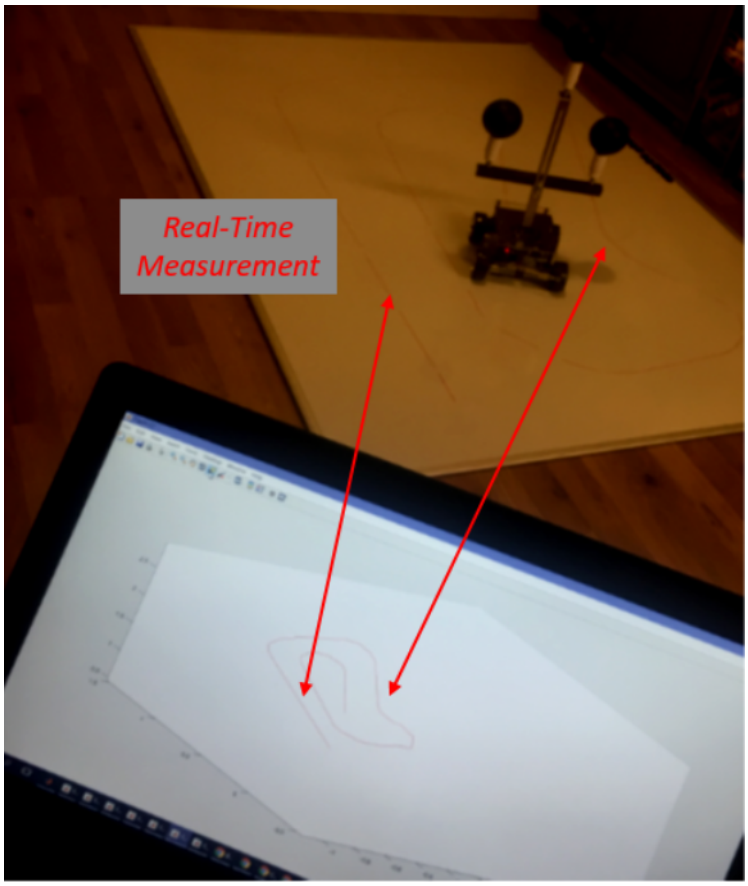


Figure 5.12: Real-Time 3D position capturing.

After capturing hundreds of samples from the camera sensor, we capture a picture of the white board from the top. We then input this picture and the camera data to the system again.

Figure 5.13 illustrates an image of the ground truth marked on the board taken from the top. Since this image is taken from the top and the camera sensor is not parallel to the benchmark board, the projection of the board onto the image

plane is most likely to be oblique projection. Extracting the ground truth from this image requires processing the image and rectifying the image plane. For this reason prior to taking the picture of the board, we install four control points on the board that are used as reference for image rectification. The process steps of rectification are explained here:

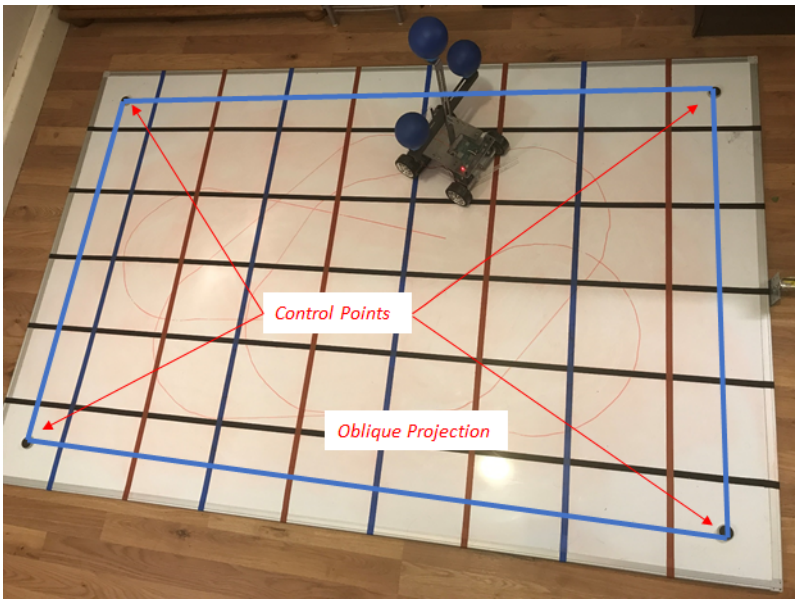


Figure 5.13: Oblique projection of the board with respect to the image plane.

(1) Detecting the control points on the image. The control points form the base of the projected plane since we know the real distances between these control points.

(2) Extracting the transformation matrix that maps the control points to new rectified points.

- To infer the spatial transformation we look at some aspects in image. Notice first that what we are dealing with is an image to image transformation so our spatial transformation is a 2D to 2D transformation. Since the scene appears tilted, straight lines remain straight but parallel lines are no longer parallel, but in fact they converge to a vanishing point. We can then deduce that we are dealing with projective transformation.
- A projective transformation is an affine transformation, adding to it a projective warp. It makes up in total 8 unknown parameters as shown in eq

5.1.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.1}$$

- Solving this equation with 8 unknown parameters requires having a minimum of 4 points. This is the reason why we use here 4 control points

(3) The last step is to map all the points in the original image to new points in the new image, using the transformation matrix calculated from the four control points. Figure 5.14

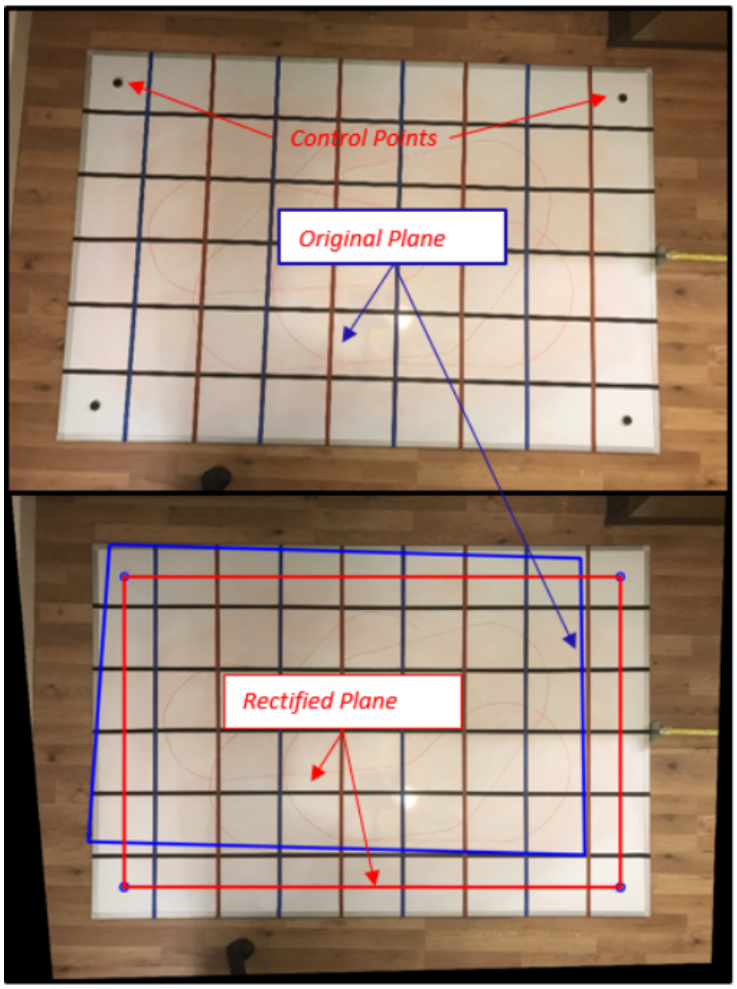


Figure 5.14: Image rectification using four control points.

In the next section we will explore addition hardware and setup used during the experimentation.

2.4 Hardware Designs and Architecture

As we mentioned earlier, we developed several versions of markers, mobile robots and benchmarks. In this section we present additional notes on the design of the markers and present a global architecture used that relates the camera sensor with the mobile robot in one entity. We proceed by exploring the global architecture.

The global architecture of the hardware components employed is illustrated in figure 5.15. As shown by the figure, our main embedded PC is a raspberry pi on which most of the peripheral processing is performed with a little bit of image processing. For heavy image processing we established a remote connectivity to a server PC for execution. The communication protocols between different components are also pointed out in the figure.

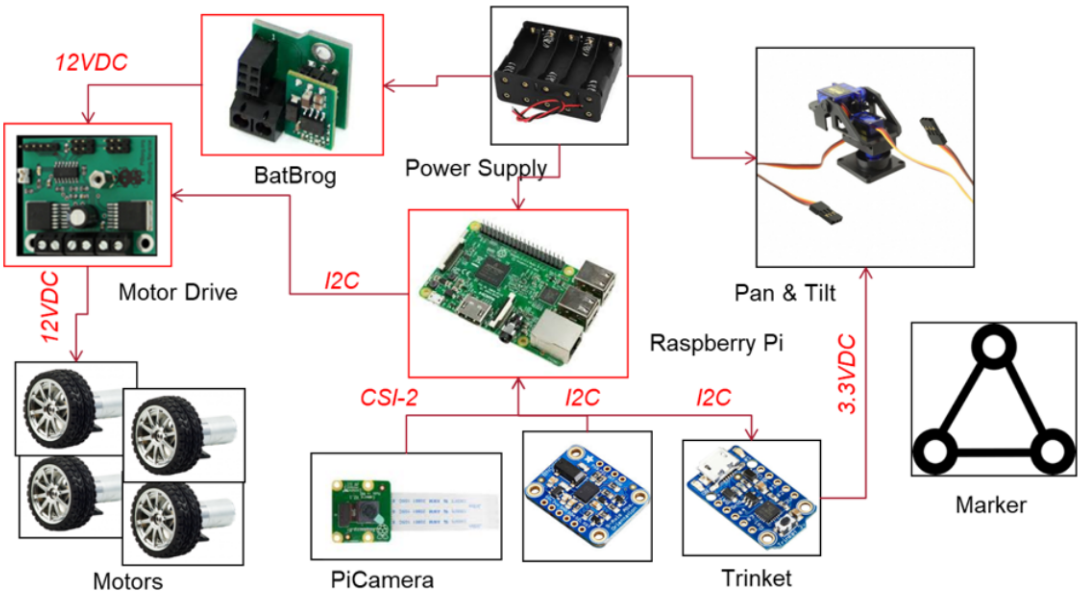


Figure 5.15: The main hardware components used around the Embedded PC.

For controlling the mobile robot, we use a differential drive concept for smooth rotation and center spinning. A motor drive chip is employed and connected to I/Os of the raspberry pi which allows controlling the speed of the motors from every side, which in turns creates rotation and straight movement of the vehicle. We also integrated additional sensor for the purpose of calibration. The IMU

BNO055 from Bosch was integrated and that provides the camera sensor with orientation data. A trinket board is added for controlling the camera pan and tilt angle if necessary. This architecture was made flexible since not all the components were used all the times. It is designed to fit in many experimental work, and is also used for capturing camera sensor and driving the robots.

To put the markers in motion, we designed and implemented three version of the mobile robots. These robots consist of embedded PC, motor drive and a camera sensor with an outfit for carrying the markers following the same architecture in figure 5.15. We also equipped the vehicle with a damping system in order to insure that the power of the motors is fully transferred into motion while providing smooth movement with full damping on uneven surfaces. Figure 5.16

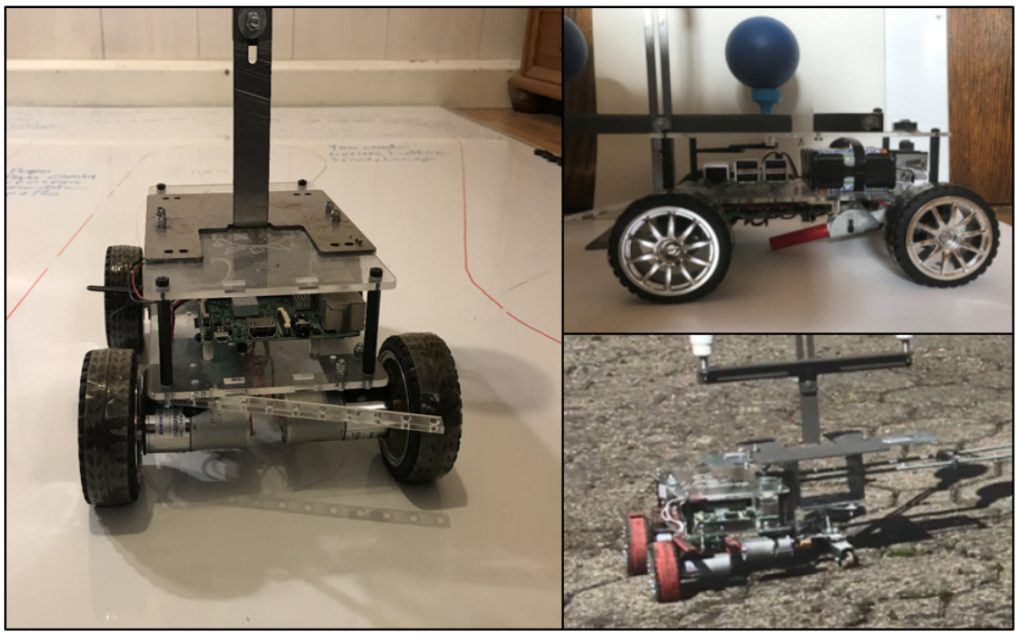


Figure 5.16: Mobile robots for holding the markers.

Testing the accuracy of the our algorithm requires efforts in processing images for good and reliable markers detection in images. For this purpose we built marker that can be easily detected in images under light changing conditions and harsh environment. In figure 5.17 we show different markers shapes used with different sizes and colors.

(a) Different Markers used during experimentation.

(b) Markers with colored LED bulbs

Figure 5.17: Images of different markers designed.

We note here first that, following the main use case of the project, which is outdoor localisation for mobile robots, we developed markers that are relevant to such application. Second, because we are using a small low resolution camera and improving image processing is not our goal in this thesis, we built markers that are not ultimately small in size ($\rho = 6$ to 12cm), with spherical vertices for the ease of the work and for accurate detection in images.

2.5 Software Architecture

Software architecture is the main building block for our experimentation. We design a modular architecture that is flexible to fit for all test runs. We will show here a quick overview of the global architecture and discuss some main software components and used open-source libraries. We will first explore the software main components illustrated in figure 5.18.

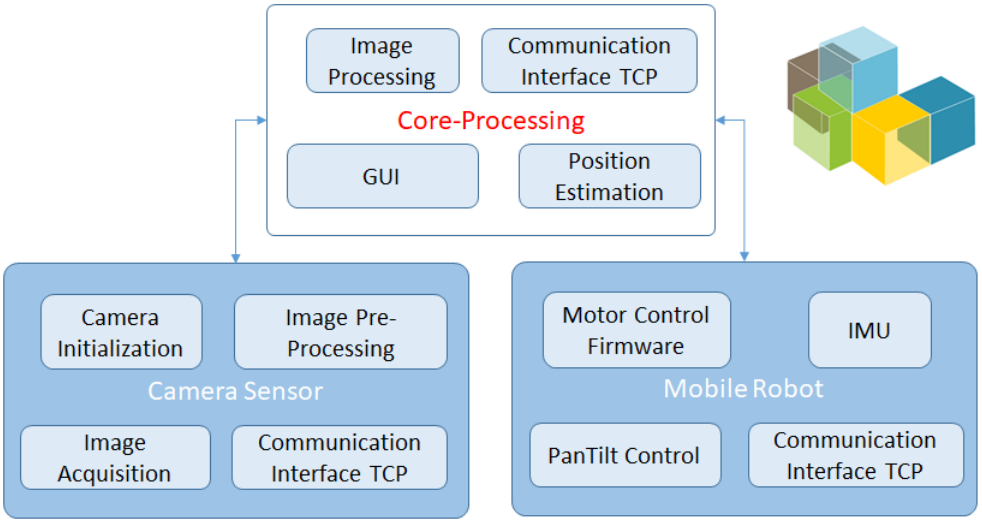


Figure 5.18: Software Components.

The software has three main modules that communicate between themselves to exchange data, which are the core-processing, the camera sensor and the mobile robot. Each component runs on separate thread. The core-processing manage the communications and main processing of the data.

A high level architecture is shown in figure 5.19. We rely on this architecture

mainly in all the developments.

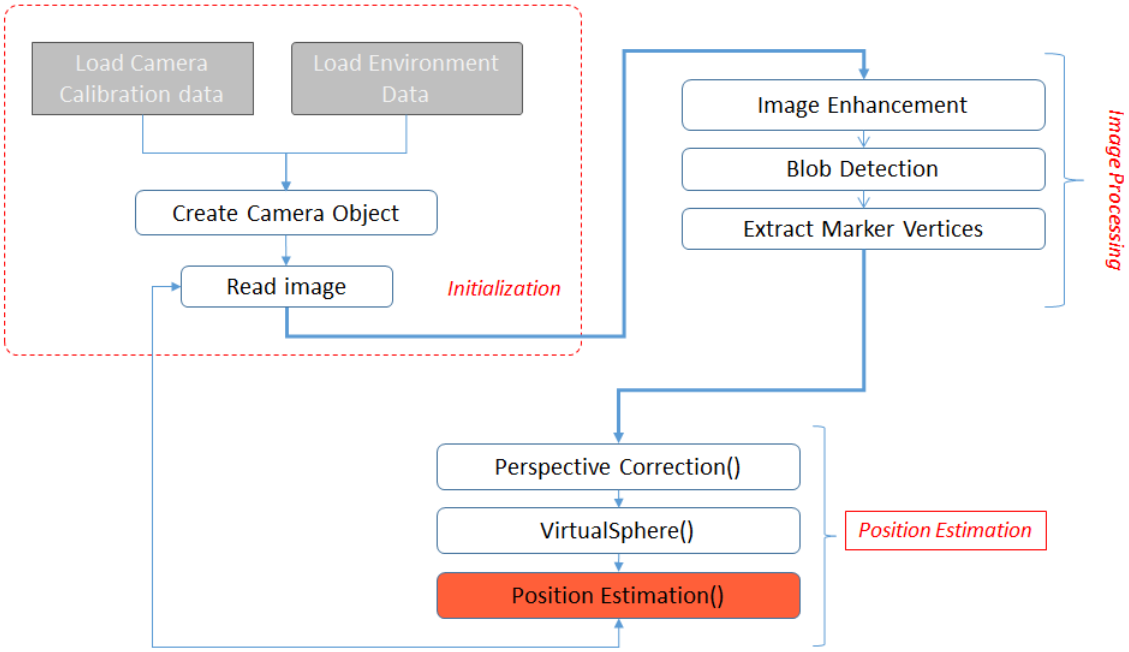


Figure 5.19: Software Architecture.

3 Experimenting the Marker in Real-World

In this section we explore couple of the experimentation we did for further proving the concept. Below is the list of experiments we wish to cover here:

- Experiment A - VS Computer Simulation
- Experiment B - Experiments on Spherical Targets
- Experiment C - VS on Circular Platform
- Experiment D - VS on Grid-Benchmark Platform

We will cover every experiment in details in a separate section.

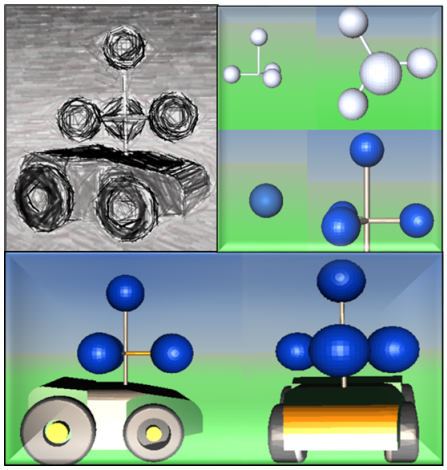


Figure 5.20: Exp. A - Markers in 3D environment.

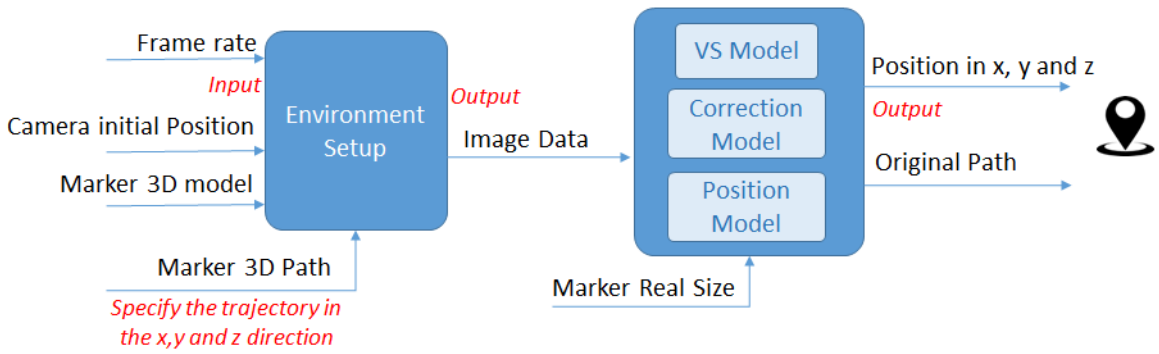


Figure 5.21: VS simulation workflow.

3.1 Exp. A - VS Computer Simulation for Position Estimation

This is the first experiment close to reality that we carried out for testing the results of the VS model. The workflow diagram is shown in figure 5.21.

In this experiment we rendered a 3D marker model along with 3D environment using VRML linked with Matlab, where the processing took place. A typical pinhole camera model is used to capture images of the scene. The intrinsic parameters of the camera are set according to our need and to the processed environment, and are fed into the model. We placed the marker within the field of view of the modeled camera. We then start navigating the marker in sinusoidal manner in X and Y-direction so that the marker is subject to foreshortening effect. While swinging sinusoidaly the marker in the X and Y-direction, we add a constant and linear motion in the Z-direction.

After setting up the environment, we run the model to capture sampled images from the simulated camera at constant frame rate. These images are first processed to detect the vertices of the marker in the image and are then fed into the VS model, followed by the correction model and position extraction model. The correction model takes care of the foreshortening and distance effect, seen in the earlier chapter, by applying center of gravity projective rotation to all the three vertices.

We take time samples from every direction of the captured data and plot out the results as shown in figure 5.22.

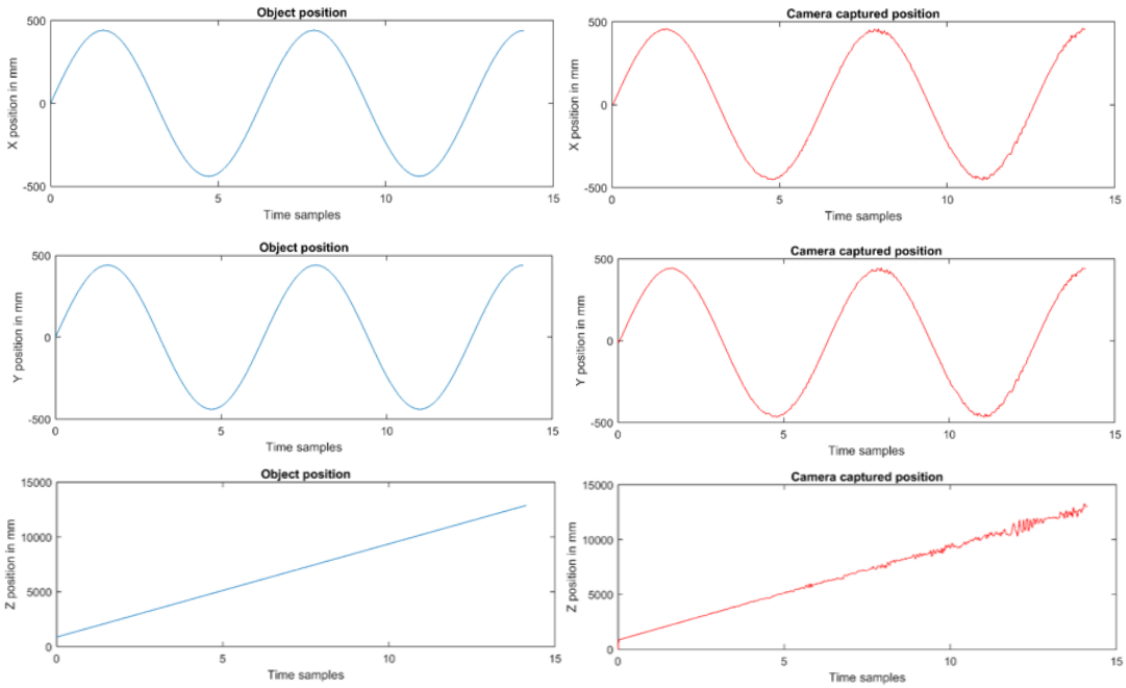


Figure 5.22: Exp. A - The left plots show the true values while the plots on the right show in the estimated data from the simulated camera.

3.1.1 Exp. A - Summary and Conclusion

We concluded the follow from the simulation:

- Since this is a pure pinhole model with a specific resolution and with no lens distortion, two noise sources are induced which are the detection process of the vertices centers and the accuracy of the correction methods.
- Looking at the results shown in the plot, we see that after 8 meters some fluctuation and noise start to appear. These fluctuations come from detection methods of the centers (Hough-Transform) and the image enhancement process.
- As our correction method relies on detecting the Center-Of-Gravity of the projected vertices, error can be induced in the correction which as we see is almost negligible.
- A big part of the noise can be controlled here by adjusting the marker size or the camera resolution.

3.2 Exp. B - Experiments on Spherical Target

The real first experiment was on spherical targets. We used the linear and circular test benchmarks to evaluate the overall performance on such targets, as we show in figure 5.23. We mainly intended here to test the accuracy of the distance effect correction providing that we have the ground truth.

Figure 5.23: These images show experimental work on spherical target

We placed first the spherical marker at different distance using the linear test bench. For evaluating the position under distance effect, we had to extract the radius of sphere in the images while we move the target away. As in all the experiments, we used the Hough-Transform technique with some image enhancement processing to try to extract the center and the radius of the sphere in the image. We then use this radius to calculate the distance of the target to the camera sensor. The plot of the distance distortion factor is shown in figure 5.24. Since foreshortening and pose effects are negligeable, the only source of distortion is from distance effect. Knowing the model of the distance effect, we can apply a correction to the extracted distances.

On the other hand, we use our Circular-Benchmark to evaluate the foreshortening effect this time and not the pose effect. The reason is that spherical target

are not affected by the pose effect due to its symmetrical shape as we already explained in chapter 3.

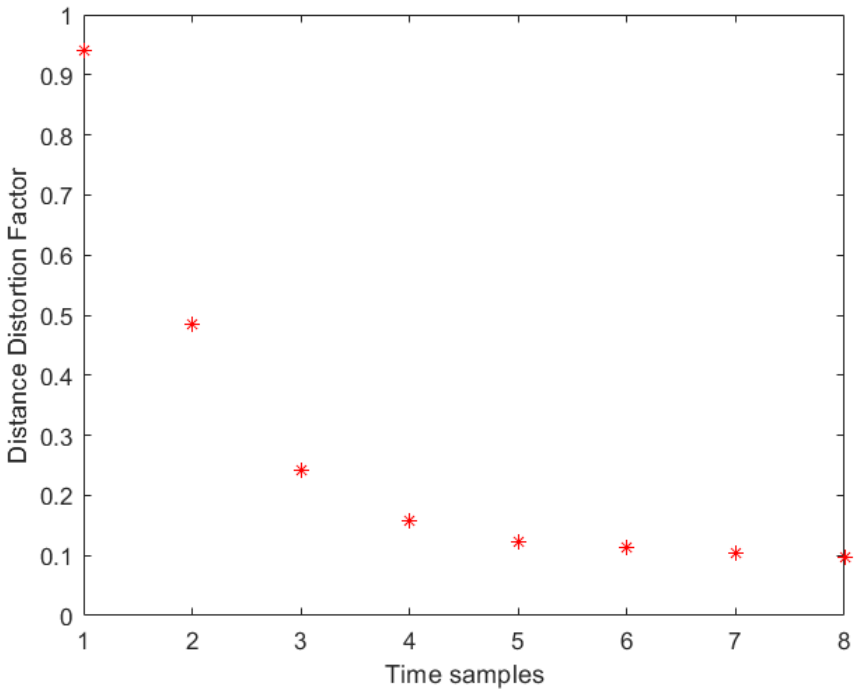


Figure 5.24: Distance Distortion Factor on spherical target.

From this experiment we gathered too much data that were most of the times too noisy. We plot in figure 5.25 some of the results. Since spherical targets are not affected by pose effect, the only distortion factor affecting our position extraction are coming from foreshortening and distance effect.

3.2.1 Exp. B - Discussions and Conclusion

During these experimentation we faced too many challenges in processing image for detecting the target. We tested different color spheres with different background in order to filter out the noise surrounding target within the image.

Since camera collects light at an angle, the area of the scene increases with distance. As a result, as the target moves away from the camera sensor while its

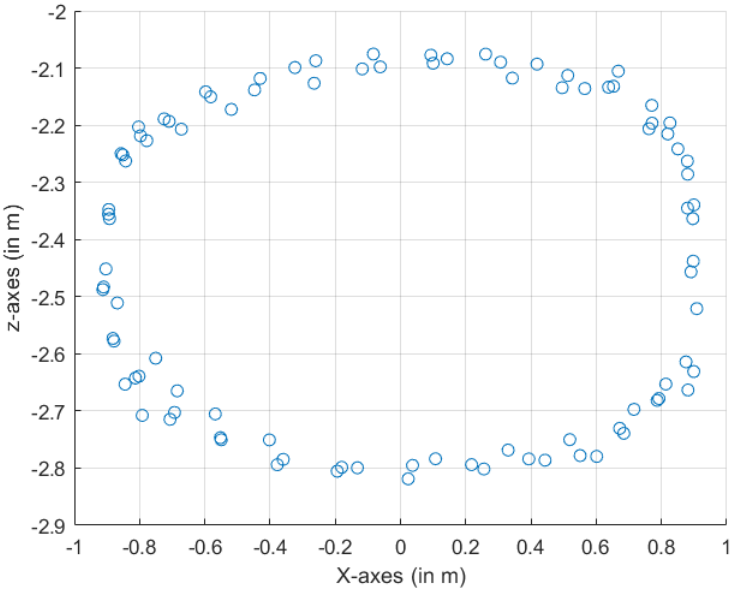
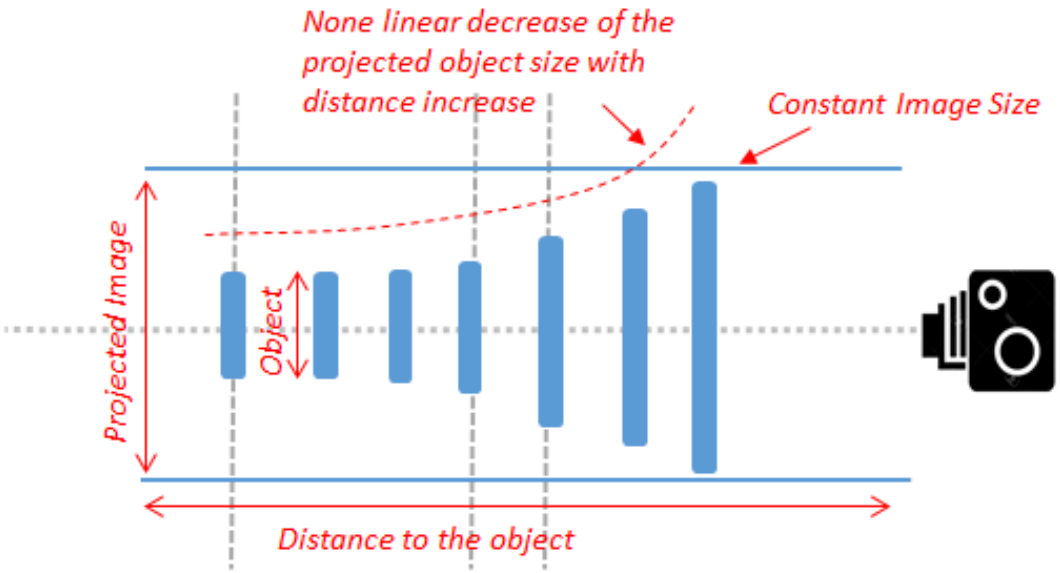


Figure 5.25: Exp. B - Noisy data from the position captured of a spherical target.

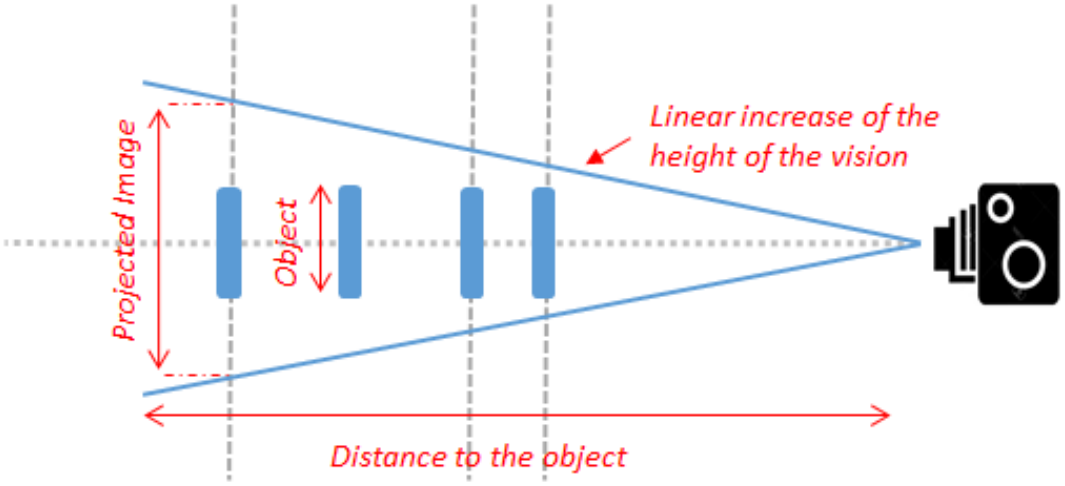
size doesn't change in the real world, the area of the object being captured by the lens becomes smaller. This leads to say that the size of the object changes non-linearly with distance. We sketch this basic fact in a diagram to point out at one bad property of such targets, as compared to other marker. The diagram is shown in figure 5.26.

The diagram shows that the further the target moves from the camera sensor, the more the projection of the object becomes similar. Considering what we mentioned here, and knowing that we are dealing with digitized images, and as object moves away, there will be not enough pixels to represent the size of that object in the image when the object is placed a little bit far. That means, with low resolution camera, accuracy drops drastically after 2 meters of distance. This fact is also shown in the results we achieved in figure 5.25. The problem becomes more complicated when trying to guess the radius of noisy ball in the image. With the VS Marker, this effect has less impact.

Our **conclusion** out of spherical targets is stated here:



(a) We show here the projected object size with respect to distance change depicted by the none linear curve.



(b) The figure shows the linear increase of the real image size projected with respect to distance.

Figure 5.26: Figure a) and b) show the proportion of the object size with the image size as the target moves away from the focal point.

- A main point to discuss here is that estimating the radius of the sphere from three points on the sphere is a much robust task than guessing the radius by measurements on the image. This is a big advantage offered by the VS model.
- the noise using spherical target is depending on scene illumination, for example the sun position, reflection of the light on spherical marker, etc.
- In contrast, our algorithm is much better when illumination is stable but in a condition of placing the marker close to the camera sensor.

3.3 Exp. C - VS on Circular Platform

We come here to testing the VS model in real world. We start by analyzing the model in rotation motion. We place our marker on the circular platform for capturing the data. The workflow diagram is illustrated in figure 5.27.

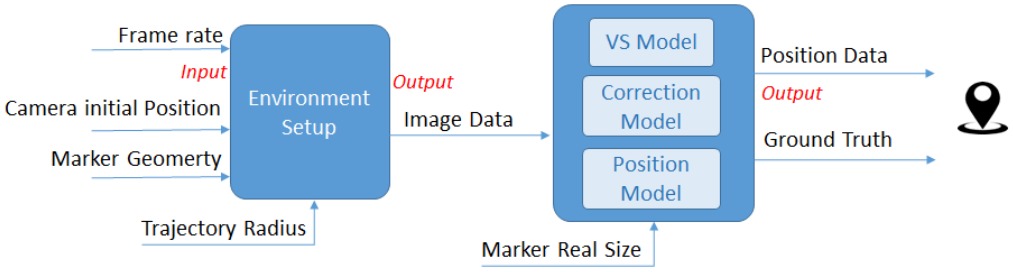


Figure 5.27: VS with Circular Benchmark Platform.

Figure 5.28: Exp. C - Test run on circular platform using the VS concept.

We perform many test runs with different configurations. Some results are shown

in figure 5.29.

In one of the experiments, we set our benchmark into operation with an equidistant marker attached, having a radius of 12cm. The radius of the circular motion is of 0.75m. For getting close to a real use case, we place the benchmark on uneven surfaces. After detecting the marker and its vertices, we perform all the perspective corrections discussed in the early chapters. The radius of the VS is then extracted, out of which we calculate the marker's relative position.

3.3.1 Exp. C - Discussions and Conclusion

Considering all the vibration from the uneven surface we can say by looking at the output that VS model is a successful model. To estimate the current accuracy, we fit an ellipse in the collected data points of the camera sensor and estimate its major and minor axis.

Our **conclusion** is stated here:

- We estimate the average error we get from several test runs to be less than 2cm at a distance of 3.7 meters from the center of benchmark.
- We have reached our goal and proven that the VS is a very robust method and is highly accurate.
- With the VS we overcome the problem of singularity when points are located in the same plane as it is the case of the marker. The radius of the marker is still detected even when the marker plane is perpendicular to the image plane.
- With the VS model, what we are interested in is to detect the vertices of the markers. This makes the system more robust when detecting the vertices instead of guessing the radius from the projection in the image.
- Improvement can be made on image processing for better detection of the vertices especially when the marker is in fast motion.

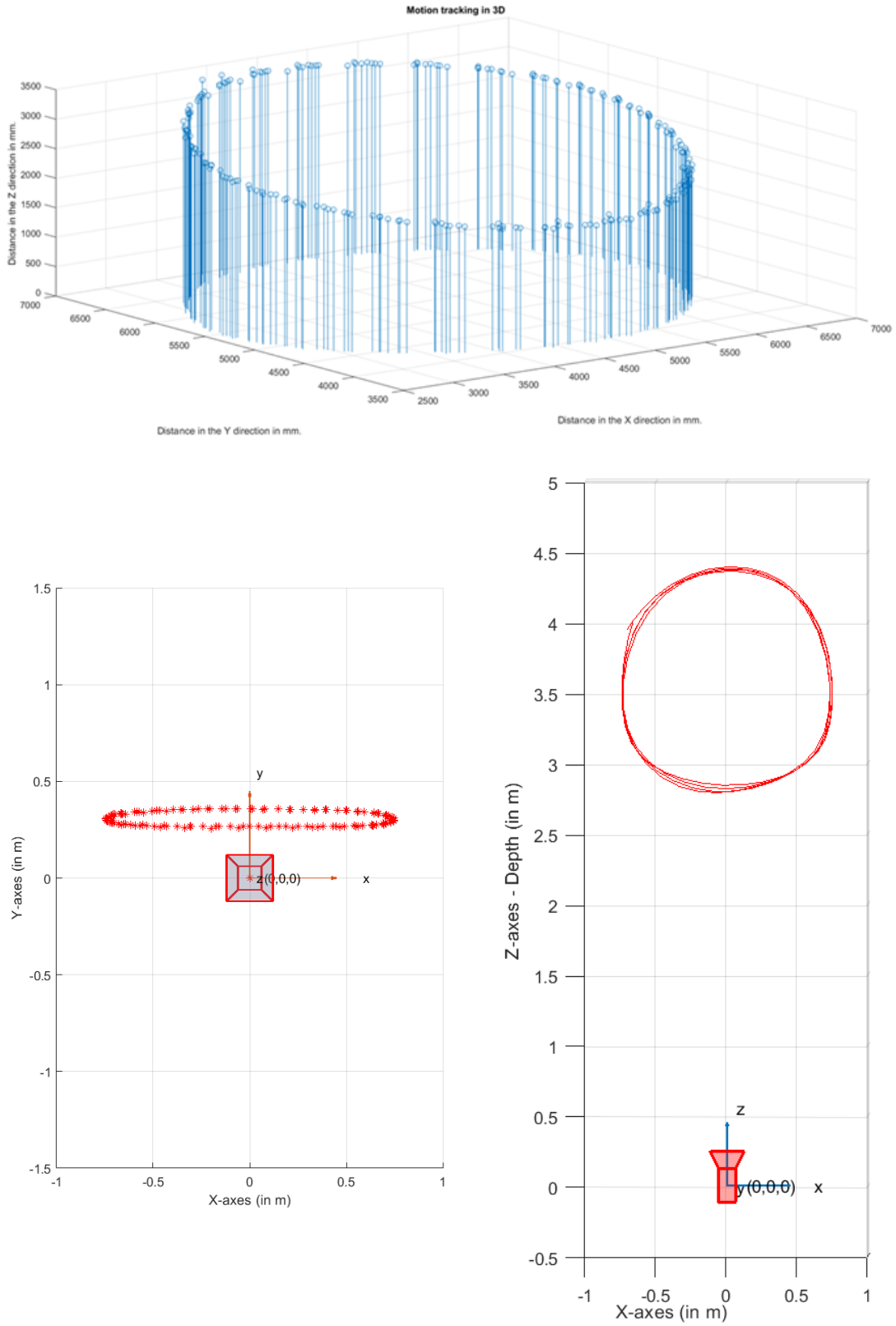


Figure 5.29: The position extracted from different experiments out of a Circular-Benchmark using the VS model.

3.4 Exp. D - VS on Grid-Benchmark Platform

We come here to a relevant experiment where the marker is no longer constrained in motion. We already explained the functionality of the Grid-Benchmark in the previous section. We detailed here the VS experiment on that benchmark. The workflow diagram is illustrated in figure 5.30

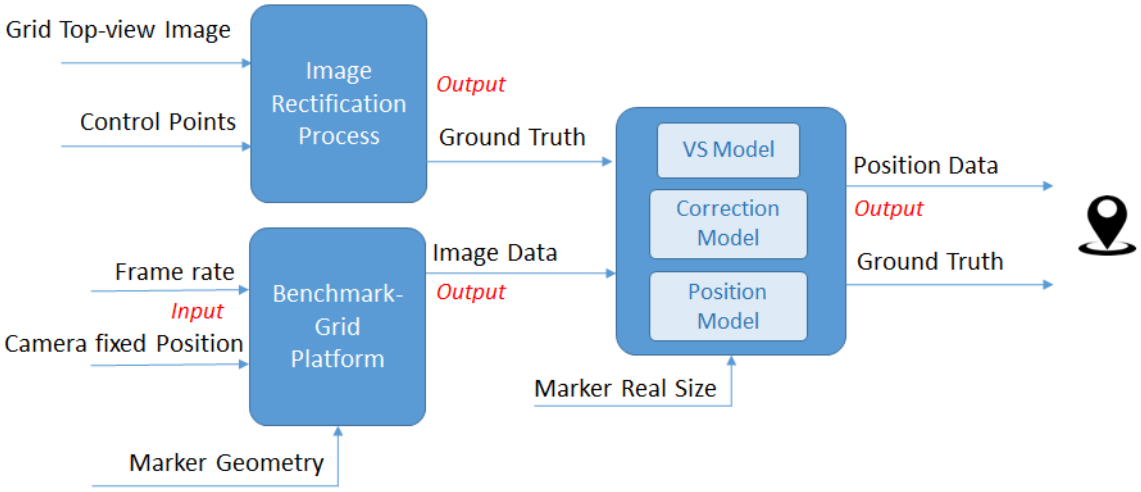


Figure 5.30: Exp. D - Workflow diagram of VS experiment on the Grid-Benchmark

We position the camera at a known position from the meshed white board. The camera orientation is measured accurately to align the focal axis with parallel plane of the board. We boot the system to start capturing data from the camera while the marker is navigating with activated marking on the board. We record the data and feed them into the VS model and then perform the perspective correction process. We use the same approach like in the computer simulation work. Once we extract the 3D position, we start collecting the ground-truth data. Figure 5.32 shows the plot of the camera sensor data.

After taking hundreds of image data from the camera sensor, we take a picture of the board from the top. The picture taken is depicted in figure 5.33. We then apply the rectification process as shown in figure 5.34.

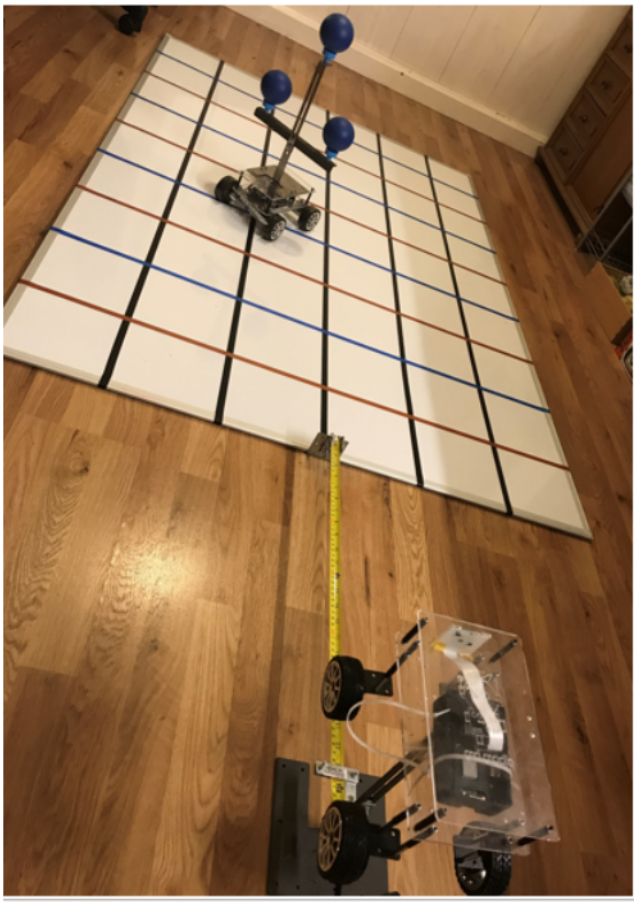


Figure 5.31: Exp. D - Triangular Marker on the Grid-Benchmark Platform.

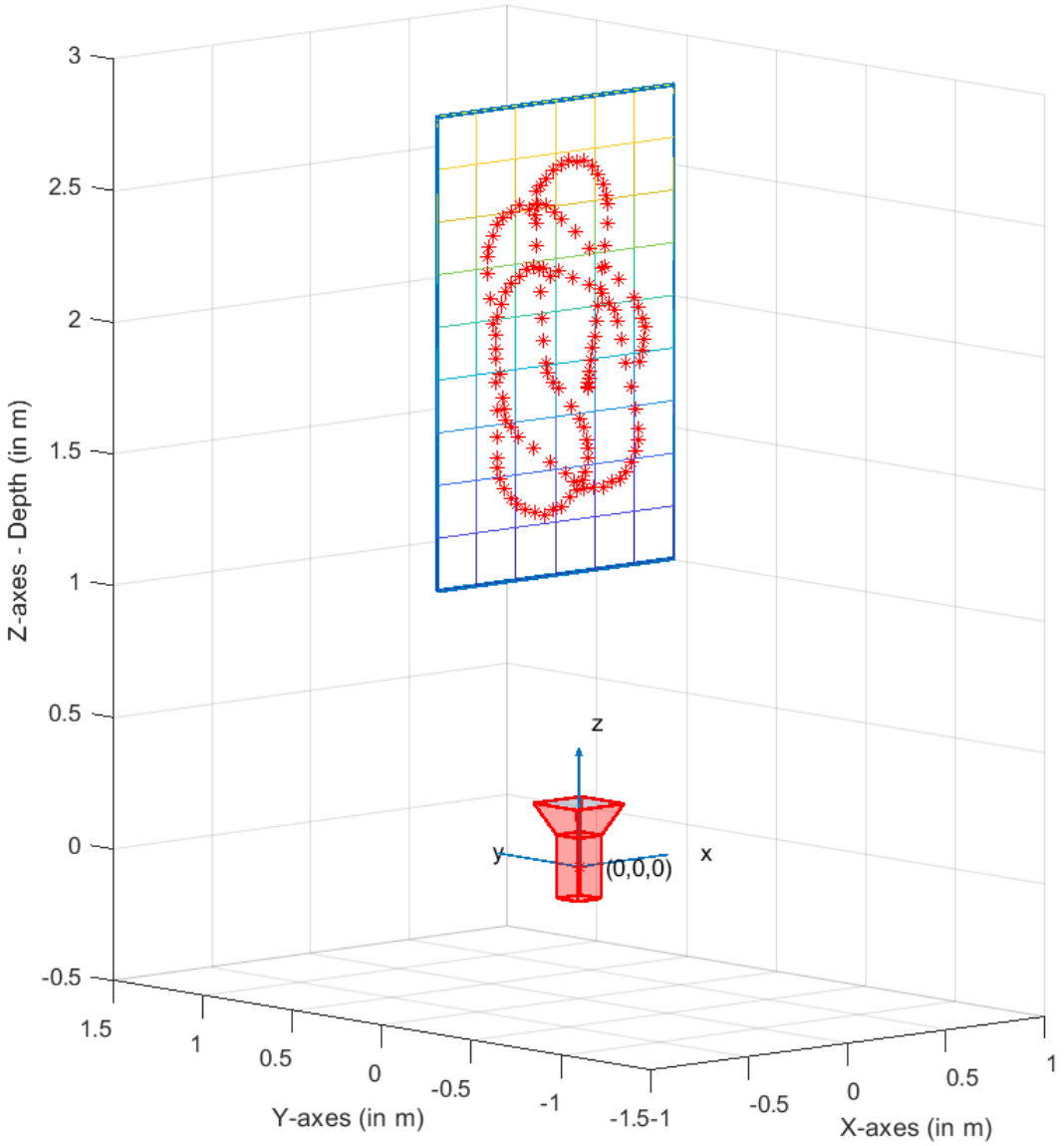


Figure 5.32: Exp. D - 3D position estimation of the maker on the Grid-Benchmark.

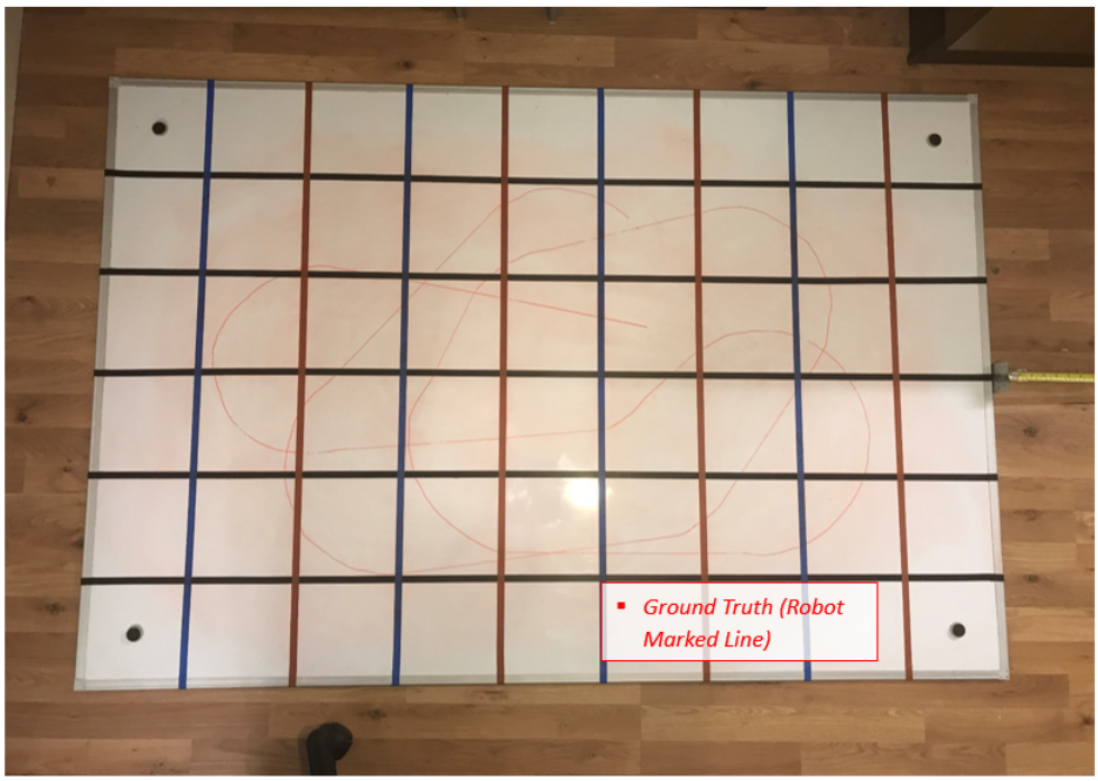


Figure 5.33: Exp. D - Ground Truth path of the maker on the Grid-Benchmark.

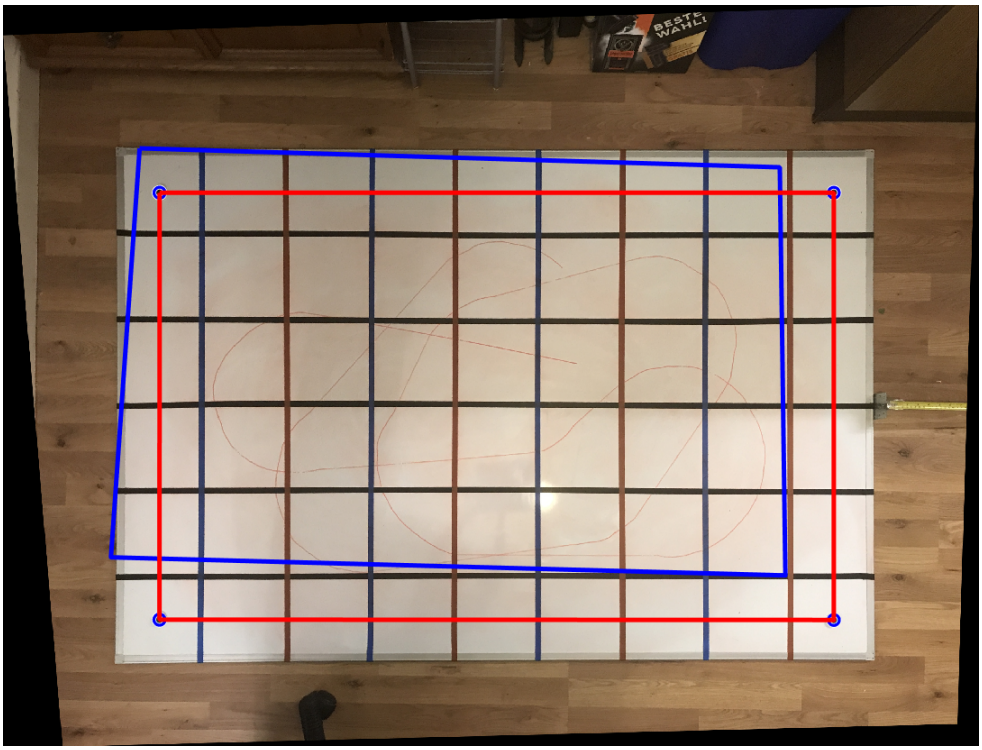


Figure 5.34: Exp. D - Ground Truth path of the maker is rectified.

3.4.1 Exp. D - Discussions and Conclusion

We can now plot and compare the results with the ground truth. We show in figure 5.35 the plot of the camera data on top of the rectified marked trajectory.

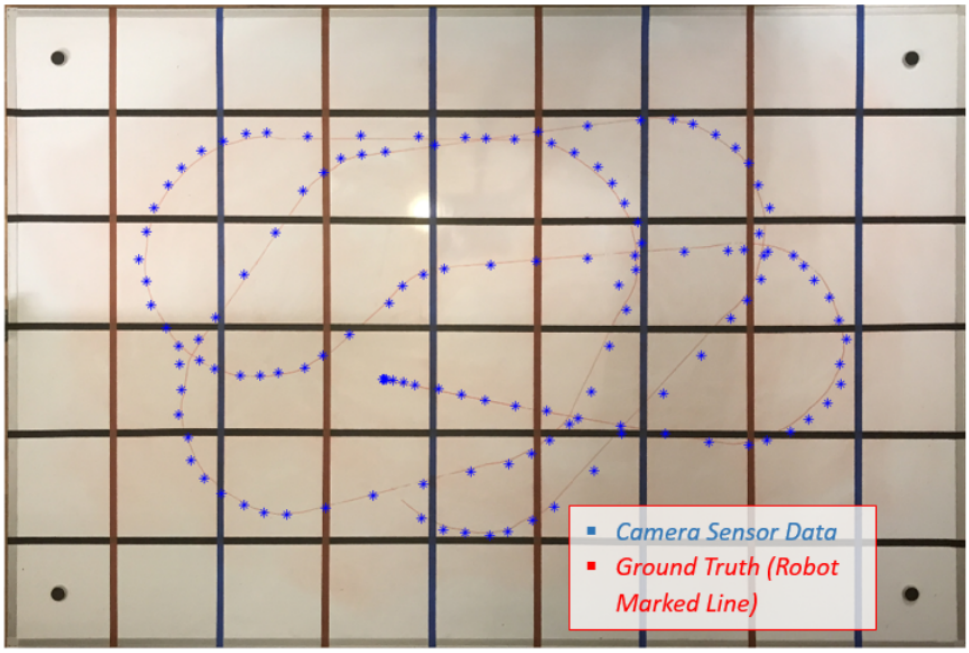


Figure 5.35: Exp. D - Camera sensor data in blue plotted on top of the ground truth marked line in red.

We state our **conclusion** here:

- In this experiment, we have successfully reached the desired goal. We have proven the feasibility of the study by experimental proof.
- We have shown that by applying the perspective correction we developed to the VS model, mainly the error in position estimation becomes negligible.
- In comparison of the VS model for the triangular marker with the spherical target, the VS model can tolerate much less noisy sources.
- We show in this experiment that further accuracy can be reached. From this benchmark we can say that the system can reach an accuracy below centimeters.
- Furthermore we can say that the precision of system is within millimeters.

- By showing these results we open the door to the elaboration of a new and simple technique for position estimation. This technique can have a wide range application in indoor position and outdoor positioning.

4 Summary and Discussion

We come to the end of this chapter. To summarize, in this chapter, we intended to prove the correctness of the concepts we developed on perspective correction and VS modeling by means of experimental work.

We developed three benchmarks each of which is designated to simulate a type of distortion in different scenarios. A Linear Benchmark Platform to evaluate the distance effect. A Circular Benchmark Platform to analyse the effect of rotation and circular motion of the marker, on position extraction. A Grid-Benchmark Platform to simulate and study the position extraction when the marker is moving freely.

As in chapter 4, we integrated the perspective correction and the VS models for relative position calculation. We used different types of markers to simulate different environments where the markers are subject to light changing conditions and vibrations due to the navigation in uneven terrains. We also used a different set of cameras and lens which added more uncertainty on the intrinsic camera parameters and provoke the extrinsic parameter to become more sensitive to environmental changes and sometimes less robust in some particular scenarios.

In analogy with the theoretical study, the results show with the help of the benchmarks that the VS model applied on triangular markers, in general works perfectly well as compared to spherical targets as shown in figure 5.25. One of the big advantages of the VS is that mainly for all the marker's positions, the system doesn't fall into singularity. This makes the model unique and more robust in terms of accuracy and precision.

Chapter 6

Scale for Monocular SLAM

Contents

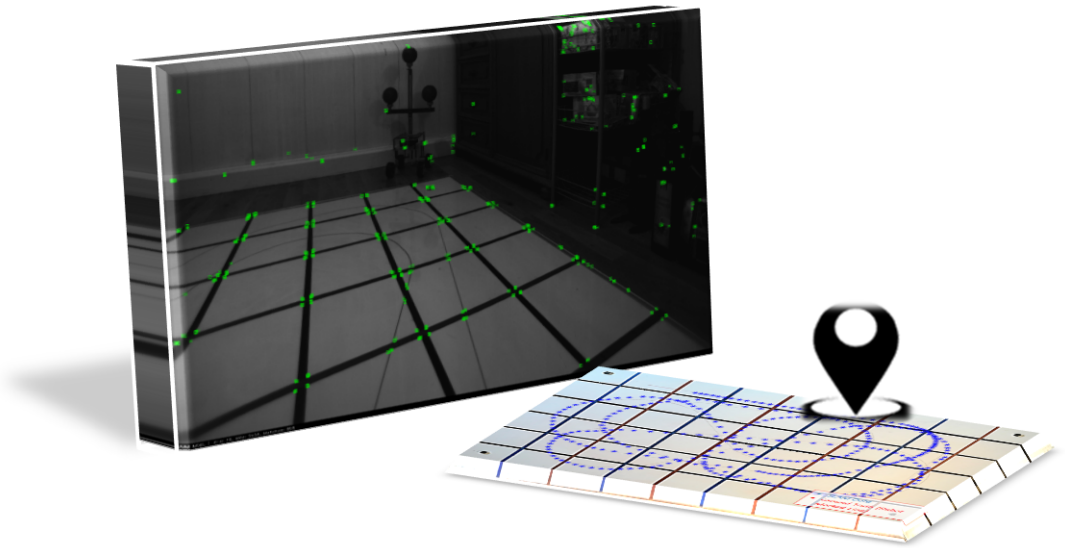


Figure 6.1: Marker-Based SLAM.

1 Chapter Summary

Monocular vision systems suffer from the lack of true scale(74). Depth is not observable from a single camera. The scale defines the relationship between the real world metric map and the estimated geometry unit map. Although ideally, this scale should be constant (119), it has been realized that monocular SLAM algorithms are prone to scale drift over time, creating different scale in different parts of the map (87), (126).

We presented the state-of-the-art of Visual SLAM in chapter 2. We recall here some approaches that mitigate SLAM scale. One solution which is presented in (119) and (113), relies on depth from defocus (DfD) to relate distance to the scene. In (119), the author compared his work presented to the state-of-the-art ORB-SLAM. Another approach is to impose geometrical constraints from the scene, like objects shapes and sizes (15), (48). We will focus in this work on solution with geometric constraints and integrate our method into the open source ORB-SLAM. There are of course other solutions to resolve scale drift which use stereo or RGB-D cameras as in (41), (97). These type of sensors are not the case of our study. Our focus is on monocular cameras solution which are attractive and versatile because of their small size, low power and low cost (19).

ORB-SLAM is the State-of-the-art of monocular SLAM systems (89), (90). It is very famous for its ability to reuse the map, detect loop closure, and perform global optimization to minimize the accumulated scale drift (119). However, it is very hard to get the right scale and if the scale is correct it is very likely to drift quickly, specially when the camera is turning (90).

A contribution in marker-based localisation is presented in this chapter. We explore a solution to calculate the SLAM scale using external Marker. The approach for scale calculation is explored in the next section.

We will start by presenting the workflow diagram as in every chapter; we then present a short theoretical demonstration followed by real experimentation using an open-source monocular SLAM platform.

1.1 Approach for Scale Calculation

We will start first by showing the short diagram depicting the work flow diagram. The diagram is depicted in figure 6.2. We show here a high level diagram of the scale estimation after capturing full frame images. We append to every image frame the time stamp as chunk. The frame is then sent to both the SLAM and the Marker-Based localisation module. Whenever the marker is detected in the image, we extract the relative position from the marker and its corresponding SLAM pose frame.

Calculating the scale requires that we acquire two different frames taken from two different positions. Once we have these two frames, the scale is calculated from the ratio of the distances generated in the SLAM world and real world Marker-Based position. The distance ratio r is depicted in equation 6.1. Let P_1 and P_2 represent the two positions vectors from two different poses in two different worlds. We append the letter w to denote the position in the world frame and the letter s to denote the SLAM world. Let $S = [S_x, S_y, S_z]^T$ represent the scale vector.

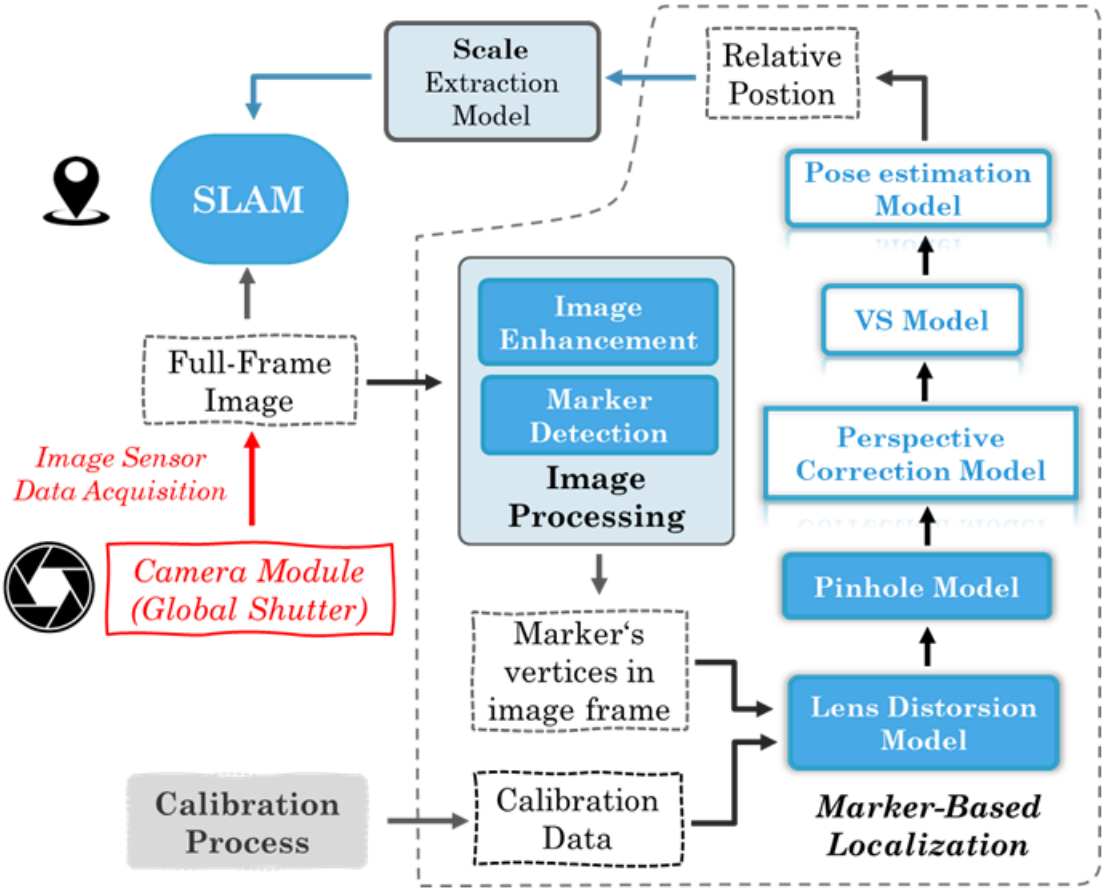


Figure 6.2: Marker-Baed SLAM Approach

$$r = \frac{(P_{1,w} - P_{2,w})^2}{(P_{1,s} - P_{2,s})^2} \tag{6.1}$$

Since the distances are linear and proportional in all directions, the scale matrix s becomes as in equation 6.2

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.2}$$

After deriving the scale, we integrate the scale into the extrinsic transformation matrix of the camera pose in SLAM world using equation 6.3

$$T = \begin{bmatrix} R_s S & t_s \\ 0 & 1 \end{bmatrix} \tag{6.3}$$

2 Open-Source SLAM

We use an open-source SLAM called ORBSLAM 2. ORBSLAM2 is a real-time open source SLAM library for Monocular, Stereo and RGB-D cameras for computing the camera trajectory in a sparse 3D map. (107).

We list here the three main components: tracking, local mapping and loop closing shown in figure 6.3.

Initializing the map is performed at the beginning. Once an initial map exists, the tracking estimates the camera pose with every incoming frame. Now that the camera pose has been estimated, local mapping is performed (8), which means, map points within a local area of the current pose are bundle adjusted to project

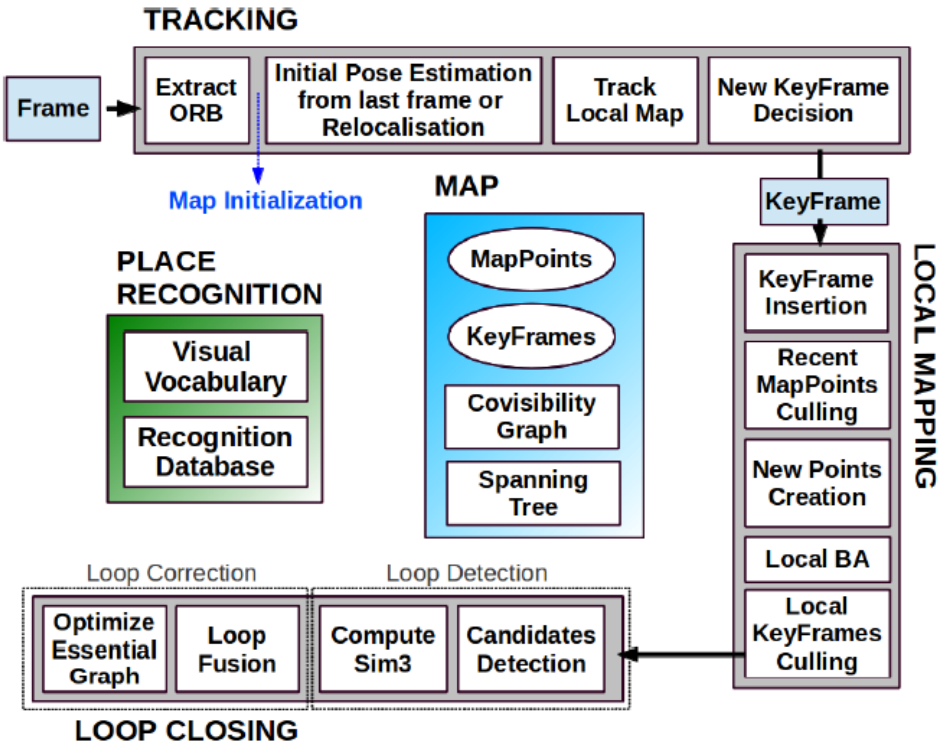


Figure 6.3: ORB-SLAM Workflow
This diagram is taken from the (89) and it shows the steps performed by three main components.

a local map (78). The use of local maps has been shown to be better suited for large scale mapping by reducing the computational cost and improving map consistency. Finally, the loop closing component serves as loop detector in the map. When a closed loop is found, a global optimization is performed to maintain the global consistency of the map.

These three components listed in the previous paragraph are the basic elements of ORB-SLAM that made it a reliable, fast and accurate SLAM compared to others. ORB-SLAM rely on ORB features for matching correspondences between two different images. These features are fast binary descriptor based on BRIEF descriptor (107), which make these features rotation invariant and resistant to noise. ORBSLAM2 (89) is a real-time open source SLAM library for Monocular, Stereo and RGB-D cameras (90), (51) for computing the camera trajectory in a sparse 3D map.

These features of ORB-SLAM made it suitable for the development of our marker based approach. Although many difficulties and challenges arise later in the development, the choice of ORB-SLAM nevertheless was successful.

3 Experimental Work

The experimental work here is to show the usability of the marker combined with ORB-SLAM. Our contribution here provides a unique method based on geometric constraint coming for the novel marker.

3.1 Experimental Setup

For this testrun we set the grid-benchmark to extract our ground truth, but this time the camera is not fixed anymore; we attach the camera to moving vehicle while the marker is placed at a random position in a way to be captured in the SLAM images. The marker-based approach concept is based on the pinhole camera model which simulates monocular vision. Although it is a pinhole model, a wide field of view lens is attached to camera sensor in real life experiment to

ensure a full visibility to the marker and for robust tracking or ORB features. Figure 6.4 shows the setup of the experiment.

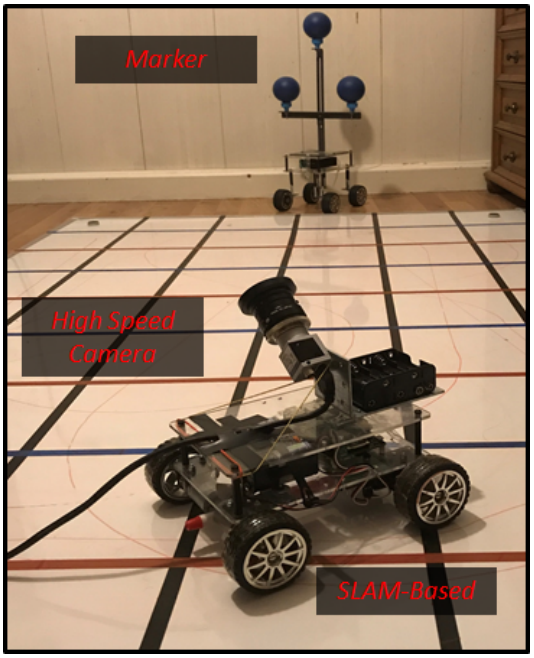


Figure 6.4: ORB-SLAM Workflow
Marker-Based SLAM Experimentation

3.2 Experimental Results

We show and compare our results in the section. The first step is to initialize the SLAM.

Initialization is shown in Figure 6.5 by the green features which are the ORB features from the ORB-SLAM. Initialization requires that sequence of images are captured from different positions. This will make sure that the RANSAC *pnp* inside SLAM has detected enough features and from different view points. Once initialized, the marker placed anywhere within the scene has to be detected. We use the hough-transform technique to detect the spherical vertices of the marker. This way we make sure we have a reliable center detection of every vertices. This are also shown in figure 6.5.

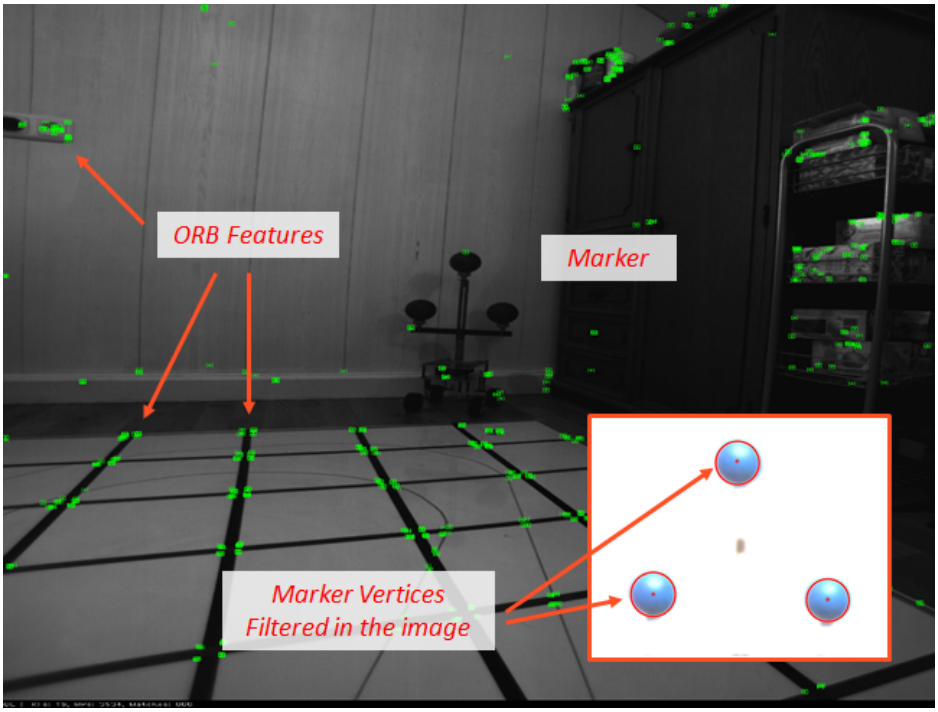


Figure 6.5: Orb feature and maker filtering and tracking in SLAM images.

Finally we take a screenshot of the ground truth, and perform the rectifications process with the help of the control points on the board, in order to plot our results as in figure 6.6. The results are shown in figure 6.7.

We note here that we added some black markers around on the walls in order to increase the ORB features and thus improve the performance of the ORB-SLAM. The results are discussed in the next section.

4 Summary and Discussions

Experimentation on SLAM was presented in this chapter. We presented our approach to solve monocular scale for SLAM using geometric constraints.

The results from ORB-SLAM compared to the ground truth illustrated in figure 6.7, show great achievements in restoring the SLAM scale perfectly. Although some drifts exist in some parts of the circuit for about 4-5cm, these are due to

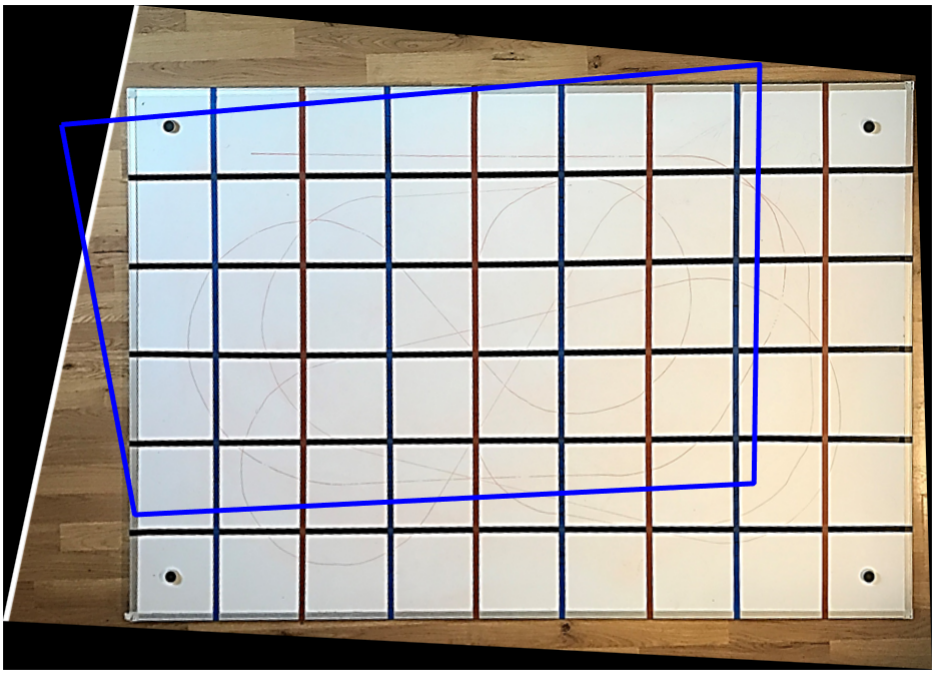


Figure 6.6: Rectified ground truth.

the lack of ORB features required for the ORB-SLAM to operate efficiently and specifically in the areas where the marker is not visible to re-calculate the scale.

We recorded a very fast scale recovery with no computational efforts required. We used an intel core i5 4th generation without any activation of GPU to load the heavy SLAM at high speed frame rate of 17 frames/sec. When monitoring the CPU, we did not encounter any CPU stress on all the cores.

Methods that use environmental and geometric constraints like (48), and which rely on camera height and configuration constraints, can never guarantee to have such fixed configuration state, specifically when moving on rough train and on uneven surfaces. Thus, accuracy could drop easily in such environment. In (15) the authors use object learning process in order to measure size of different known objects in the scene. This method can be efficient in restricted environment to known objects in addition to the big database and computational efforts required to recognize the objects and guess the correct sizes. As opposed to method

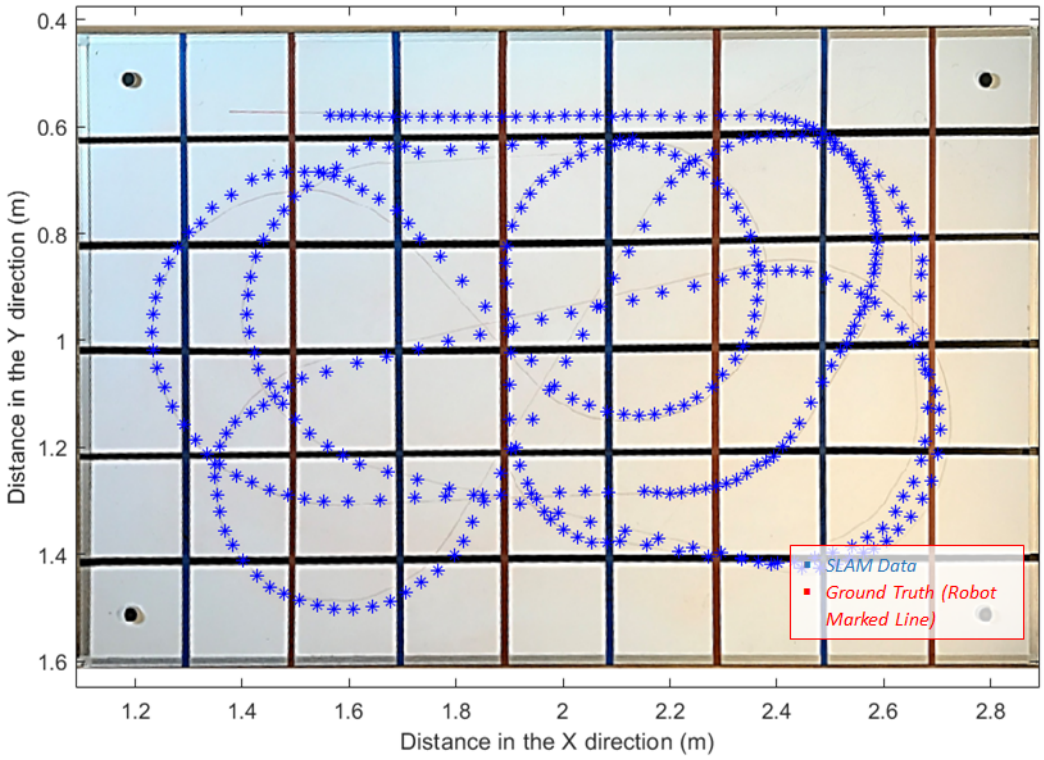


Figure 6.7: Result out of the marker-based SLAM

that uses environmental constraints like the one we listed above, the method we proposed offers many advantages that we list here:

- accurate scale calculation based on deterministic approach.
- very fast scale recovery with extremely low computation cost because of its deterministic property
- although we use a predefined geometry, we show an approach in the final chapter for making the system unrestricted to a unique geometry.
- Finally our system rely on a minimum of three points in space which makes its solution unique.

A final conclusion and discussion is presented in the last chapter.

Chapter 7

Conclusion

Contents

1 Thesis Summary

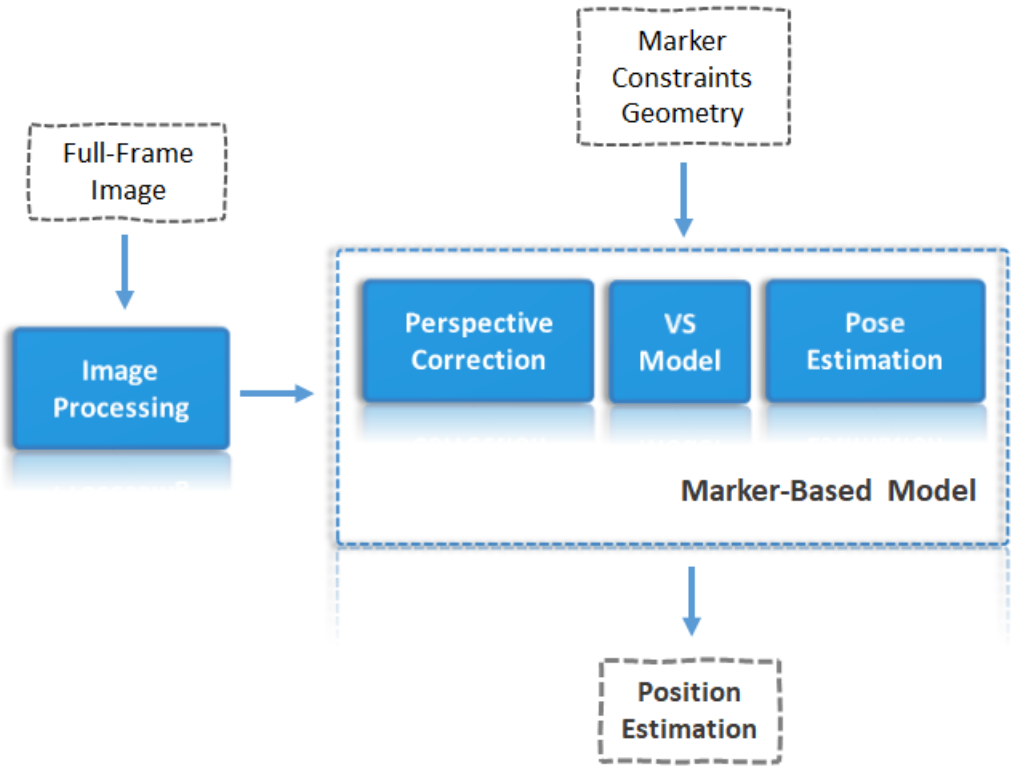


Figure 7.1: Architecture of the Marker-Based Localisation Model.

This thesis considers marker-based approach position estimation from monocular view. We emphasize on modeling perspective distortion and projective correction related to these distortions as a first stage. Modeling distortions are important in order to understand the various types and effects. Understanding the natural behavior of these distortions allow us to model and apply a correction process, that we called projective correction, in which we try to close the gap between perspective and parallel projection. On the other hand, we developed parallel projection models for triangular markers, which we use to extract the 3D relative position to the target. Furthermore, we have presented mathematical demonstrations, simulations and real time experiments to backup the proofs.

Figure 7.1 shows the simple architecture of the Marker-Based Localisation model including different modeling layers.

1.1 Chapters Summary

We started the thesis with Chapter 1 in which we presented the motivation behind the work.

In 2 we provide literature reviews and background which are relevant to the understanding of the work presented later.

In Chapter 3 we presented an overview of distortion effects from our perspective point of view, based on John Y Aloimonos's description in (6). Perspective effect modelling in its three forms were then presented followed by examples on different types of image targets like line circle and spheres. Correction methods are then explored in which we used projective rotations presented as models. We end the theoretical work by practical simulation showing the output of the models on spherical targets, depicting the correction algorithms on restoring spherical eccentricity in images. We then discussed results in the summary and discussion section like in every chapter.

Chapter 4 was dedicated to exploring parallel projection models of three points in space representing the vertices of a triangular marker. The VS and VS2 are two models in which points existing on spheres are projected. The point is to consider that these are part of a spherical target and not triangular marker since we showed in chapter 3 that spherical targets are immune to distortion coming from pose effects. Although distortions exist, these are masked by the spherical shape. We presented our methodology for doing so, whereby we showed mathematical derivations and presented recovery process from perspective foreshortening distortions. We add to these models the position derivation model. We finished by providing results of computer simulation followed by discussions and summary. The results demonstrated agreed completely with our presented models showing accurate position estimation.

In chapter 5 we provided an overview on the available experimental platforms and data-sets acquired. We highlight the hardware used and explore the experimental

setup as a first stage. We then presented the three benchmarks and their relevant purpose is to point out at the effects of distortions. This was followed by the architecture employed in all the experiments. The work of this chapter presented experimental evidence of our correction algorithms. We created many testruns scenarios from which results are captured. We plotted and analyzed every experiment separately in the discussion sections. We ended every experimental section with a summary and conclusion section. We concluded that the performance of the VS combined with the perspective correction have given accurate results with regard to camera resolution and the errors in the measurement. Using our marker-based approach, we have significantly reached an accuracy below centimeter with precision in millimeters when measuring our position at ranging from 2 and 7 meters with low resolution camera. Our result shows compensation for all the three effects including pose effect, using three points independent from their position in space.

Finally in Chapter 6, and in light of the work and results that were presented earlier in the thesis, SLAM-Based localisation was considered focusing on the ORB-SLAM state-of-the-art. Different approaches for scale recovery were investigated, which exposed the importance of accounting of accurate geometric structure as reference. We presented our approach for restoring the scale based on the geometric marker modeled in the previous chapters, with distortion corrections. Result shows a very fast recovery of the scale with a minimal computational effort due to the deterministic approach developed. By solving the scale problem, the complete SLAM output was recovered in the real world frame.

1.2 The Localisation Solution and Advantages

The problem of camera pose estimation in a $3D$ world can be referred to as perspective-n-point problem, namely, the PnP problem. A review on the PnP with it various solution is detailed in (136). Our solution to solve the localisation problem compete with the solution of the $P3P$ problem.

Solutions for solving the $P3P$ are many. Yihong et al. has added the studies into two categories: the ones that study multiple solutions of non-linear problems and

the ones that study eliminations or other solving methods.

Our solution is different from the rest of the solutions from two aspects. We first considered the points in space as whole geometry and dealt with the shape of the geometry rather that considering points in coordinates space. This offers a big advantage in freeing the marker from being stuck in single position forever. The marker can move in space and the relative position can be extracted at any time, knowing the shape geometry of the marker. In the second aspect, we looked at the problem from a particular perspective. We split the pose into position and orientation and then solve each separately. This provided us with a straight forward method to calculate solution of the position and then estimate the orientation.

As a result we have a very deterministic approach for solving the localisation problem, a fast and simple implementation and a very accurate and precise results as shown in the previous chapters.

1.3 Experimental Results Discussions

We presented, in chapter 4 and 5, simulations and experimental work for the developed localisation model. We created experimental platforms and experimented on the Marker-Based approach on different scenarios which we list here:

(1) Perspective Correction and Projective Rotation - Computer Simulation (Chapter 3).

(2) Experiment A - VS Computer Simulation (Chapter 4, 5).

(3) Experiment B - Experiments on Spherical Targets (Chapter 5).

(4) Experiment C - VS on Circular Platform (Chapter 5).

(5) Experiment D - VS on Grid-Benchmark Platform(Chapter 5).

(6) SLAM Implementation (Chapter 6).

We summarize here the analysis of experimental outcome:

- In (1), a study on eliminating the eccentric error was conducted. We showed the capacity of the perspective correction and projective rotation models to reduce the eccentric error to negligible values which is crucial for correct image measurements.

- In (2), we tested the VS model with simulation. The only considered intrinsic parameter in this experiment are the focal distance and the pixel size. We performed simulation for the mathematical models of the VS concept and real world simulations with VRML platform for further proof of concept. The result of the simulation showed a complete analogy with the theoretical studies. We also show the high level of accuracy (below centimeters) within a range of 12 to 15 meters with a low resolution simulated camera setup.

- In (3), we presented real experimentation on spherical marker and compared it to the VS model. We confirmed the ability of the VS model to improve localisation of its center and thus to have much reliable and accurate localisation of the target.

- In (4), we test the VS in on the circular platform. We showed here the independence between the extraction of the position and the orientation. Once again the system succeeded to proof the reliability of the VS model and the ability to localise at more than 8 meters from low resolution camera with high accuracy in a real world experiment, considering the harshness of the environment.

- In (5), we test the VS model when the marker is moving freely in 3D space. We showed as well the high accuracy provided in different environment and different lighting conditions. We also showed the ability to extract the position from all angles and when all the markers vertices are located on a perpendicular plane to the camera image plane.

- In (6), we integrated the Marker-Based model into the SLAM architecture. In this context, we put the marker on the side, visible within the camera field of view, to initiate the SLAM with a measurement of the real scale. We showed the ability of the combined system to extract accurate position data even when the marker is not visible to the camera.

2 Contributions to the State-Of-The-Arts

In the state-of-the-art of localisation, our work contributed to accurate localisation in different indoor and outdoor environment, where accurate positioning is a necessity. Figure 7.2 shows the contribution to different state-of-arts related to the work. We provided as well in the first two chapters literature in this regard.

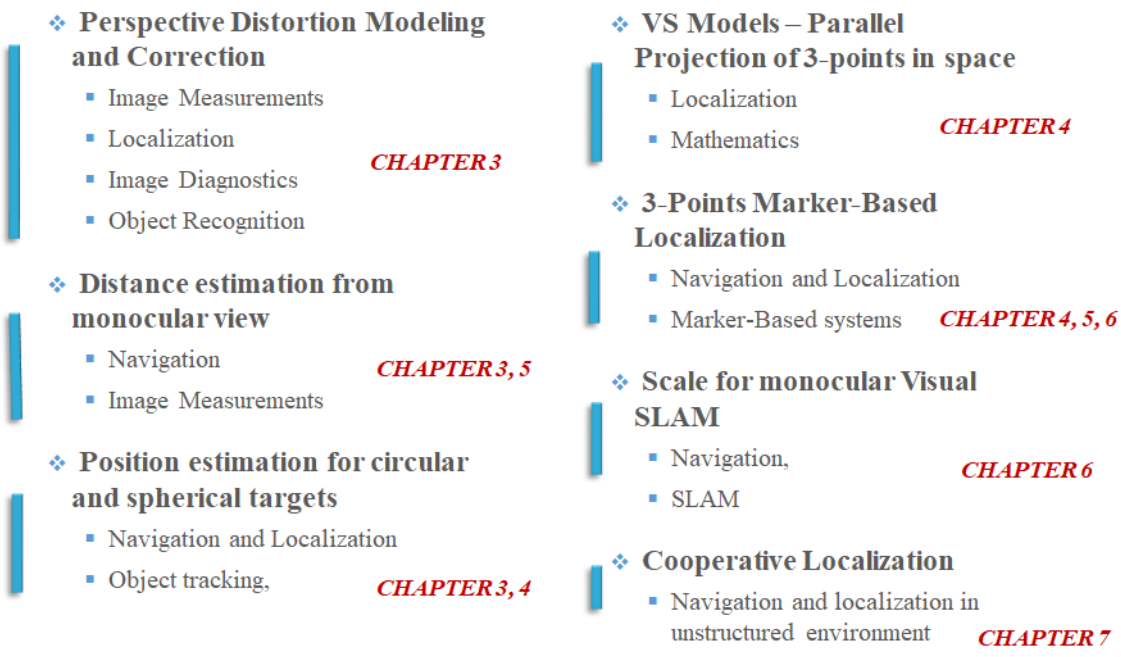


Figure 7.2: Contributions to different fields.

In the state-of-the-art of monocular vision, we dedicate the biggest part of the contributions. We list here our achieved contribution:

- A first contribution is in modeling distortions, in a unique way that can be used for correcting projected shapes in images, thus, improving its usability in localisation, object recognition and tracking.
- A second contribution in which we provide correction method for every distortion model and for different case studies and targets in images.
- A third contribution is by modeling the projection of three points in space by taking the projected distances as input and by outputting the radius

and center of the sphere into which these points are located. This model contributed in object localisation in space and distance estimation.

- A fourth contribution is in position estimation model of a 3D Object that we presented. We give full proof of a deterministic position estimation approach while we showed that it is also possible to extract the pose in closed form solution in some case.
- We finally showed a way to extract the scale out of the three points in a very efficient way compared to other methods, specially when most existing methods use *pnp*, which could fail when points are co-planar.

To sum-up in the context to the state-of-the-art, our marker-based approach is monocular vision system that requires prior knowledge of the marker geometry. The marker can be passive or active by transmitting light signal for identification. It is a map-less localisation which provides relative position but can be extended to absolute and global positioning. The marker can be used in two different scenarios. One is for object tracking and the second which is for navigation where navigation can happen in unstructured environment and can be extended to unknown environment when combined with SLAM.

3 Contributions to Indoor Positioning

In chapter 4 and 5, we presented simulations and experimental work for the developed positioning model. Results show the ability to reach high level of accuracy and precision below centimeters.

Several use cases for indoor positioning in industrial, commercial and retail environments can arise from this developed work. For example, Industry 4.0 which requires reliable and accurate localization, can truly benefit from such a system. Also, within warehouses and industrial production lines and where guided vehicles require high installation costs and lack flexibility when changing environment layouts. Another use case for such a system is where manufacturers want to abandon wireless radio technologies within their factories and environments are prone to EMF interference rendering wireless radio technologies. We mention

as well, areas and certain environments where there exists regulations on which RF-technologies cannot be used (e.g., hospitals).

We present here a summary of the current indoor localisation techniques currently in use and compare them to our system depicted in red in table 7.1 - (110), (142), (79), (141), (121), (72).(138), (10), (95), (124)

As compared to different technologies, the Marker-Based approach presented in this work can offer several advantages. We list here a summary of advantages:

- **Accuracy:** It can be much more accurate than wireless technologies such as Wi-Fi (Accuracy: 5-10m/1-25m) and Bluetooth beacons (Accuracy: 20-70m/0.50m-15m) and other listed technologies.
- **Precision:** the developed system can be very deterministic an precise unless the targeted marker is shaded. On the other side, wireless technologies can lack precision in environment with high Electromagnetic fields (EMF) interference.
- **Flexibility:** the developed system allow the targeted object to freely navigate as compared to guided wired technologies that cannot be used in environment where floor layout requires constant changes. The system can as well function in outdoor and indoor environment.
- **Sensitivity:** the developed system is Electromagnetic interference free (EMI-free). It does not interfere with radio systems as compared to wireless technologies.
- **Reliability:** the developed system is very reliable, capturing measurements cannot fail unless the target marker is outside of the acceptable range of detection or is outside of the field of view of the camera.

4 Challenges and Future Development

Although the work presented in this thesis contributes greatly to Monocular localisation, nevertheless improvements to the work can be conducted.

State-Of-The-Art of Localisation		
Technology	**Range/Acc. (m)**	**Pros and Cons**
Inertial Measurements	50-200m/0.20m-1m	• Only hybrid solutions • Requires frequent calibration • Unstable • Subject to EMF interference
Wi-Fi Beacons	5-10m/1-25m	• Signal can be blocked by walls • Frame Rate: 90fps • Subject to EMF interferences • Cannot be used for navigation
Bluetooth Beacons	20-70m/0.50m-15m	• A low cost hardware • Sub-meters accuracy • Subject to EMF Interferences • Cannot be used for navigation
RFID	0.05m-50m/0.10m-3m	• Sub-meter Accuracy • Requires Tags and readers • Short range
Guided Wire	-	• Used for navigation • Very accurate • Navigation bounded by the rail • Extremely expensive • High maintenance cost

Technology	Range/Acc. (m)	Pros and Cons
GNSS/GPS	-/0.005m-30m	• Low cost sensors are inaccurate • High cost for RTK • Require based-station • High maintenance cost • Require based Antennas in field • Can loose availability
Marker-Based (Our Approach)	1-10m/0.005m-0.06m	• Low cost sensor • Low cost maintenance • Require Marker • Can be very accurate (below cm) depending on the camera resolution • For positioning and navigation • Environmental Friendly

Table 7.1: State-Of-The-Art localisation

1. The approach we presented in this work is based on deterministic process for correction perspective distortion. A continuity to the work can be manifested by modeling a stochastic process which could open the door to further improvements in the case of unknown marker geometry and structure.

2. Scale drift and correction can also be improved in the case of SLAM. This can be done by integrating the marker's full pose within the SLAM.

3. Although we proposed the use of marker-based approach, this can be greatly extended into marker-less approach SLAM, in which we use existing scene structure to guess and find the triangular geometry within the scene. This could give great benefit to localisation in unknown environments.

4. We pointed out in chapter 1 to an approach which contributes into cooperative localisation, where robots swarms require self navigation in unknown and unstructured environments and where GPS signals cannot be reached. We recall here our concept from section 2.6 as in figure 7.3.

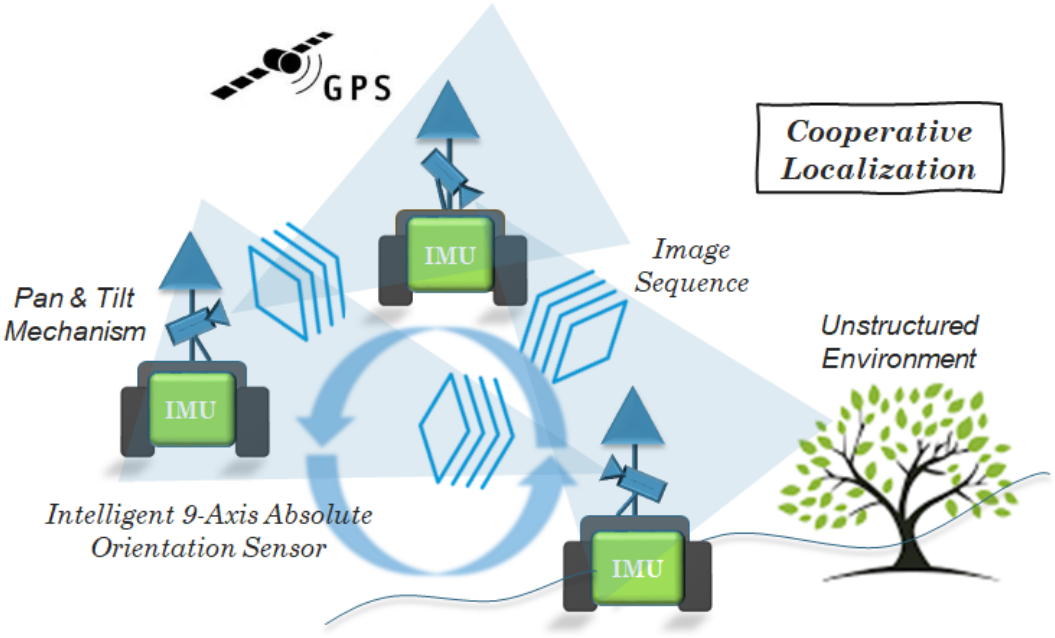


Figure 7.3: Marker-Based Cooperative localisation Concept.

Our marker based-approach could perfectly fit into this concept since we have proven to capture a very accurate relative positioning, while integra-

tion in SLAM fused with IMUs can provide additional continuity to the navigation, in the case where there is no visibility to the marker.

5 Closing remarks

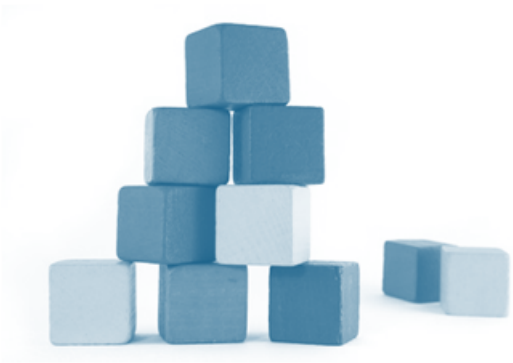


Figure 7.4: Building Blocks.

Using camera based system is still a challenging task and is still open to ongoing research. During the work, we faced so many challenges. An important challenge was to deal with frequent light changing conditions. Cameras work best in environment with stable lighting and with less reflection. Image processing for de-noise and filtering images is also challenging when working in environment with too much colored object diversity. This problem is overwhelmed in low lighting conditions. Other things to consider are the camera vibrations when moving specifically and when working with low frame rate. This could lead to motion blurring, object distortion at high level which are more recognized in still images. These are important aspects that one should carefully consider with vision systems.

On the other hand, looking at the scale of technology readiness level (TRL) we have reached the third readiness level in which the concept has been validated. We strongly believe that our work has added a building block in this era of research and development and that more blocks will come on top.

References

[1] A. Christian, J. and Cryan, S. (2013). A survey of lidar technology and its use in spacecraft relative navigation.

[2] Aalsalem, M. Y., Khan, W. Z., and Arshad, Q. A. (2012). A low cost vision based hybrid fiducial mark tracking technique for mobile industrial robots. *arXiv preprint arXiv:1210.0153*.

[3] Abidi, M. A. and Chandra, T. (1995). A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 17(5):534–538.

[4] Ahonen, T., Matas, J., He, C., and Pietikäinen, M. (2009). Rotation invariant image description with local binary pattern histogram fourier features. In *Scandinavian conference on image analysis*, pages 61–70. Springer.

[5] Ahyun Lee, Lee, S., and Choi, J. (2011). Markerless augmented reality system based on planar object tracking. In *2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–4.

[6] Aloimonos, J. Y. (1990). Perspective approximations. *Image Vision Computer*, 8(3):179–192.

[7] Aulinas, J., Petillot, Y. R., Salvi, J., and Lladó, X. (2008). The slam problem: a survey. *CCIA*, 184(1):363–371.

[8] Aulinas, J., Salvi, J., Llado, X., and Petillot, Y. (2010). Local map update for large scale slam. *Electronics Letters*, 46(8):564–566.

[9] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

[10] Barnes, J., Rizos, C., Wang, J., Small, D., Voigt, G., and Gambale, N. (2003). Locata: A new positioning technology for high precision indoor and outdoor positioning.

[11] Beevers, K. R. and Huang, W. H. (2006). Slam with sparse sensing. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2285–2290. IEEE.

[12] Ben-Afia, A., Deambrogio, L., Salos, D., Escher, A., Macabiau, C., Soulier, L., and Gay-Bellile, V. (2014). Review and classification of vision-based localisation techniques in unknown environments. *IET Radar, Sonar Navigation*, 8(9):1059–1072.

[13] Bieniawski, S., Halaas, D., and Vian, J. (2008). Micro-aerial vehicle flight in turbulent environments: Use of an indoor flight facility for rapid design and evaluation. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6512.

[14] Bonin-Font, F., Ortiz, A., and Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296.

[15] Botterill, T., Mills, S., and Green, R. (2013). Correcting scale drift by object recognition in single-camera slam. *IEEE Transactions on Cybernetics*, 43(6):1767–1780.

[16] Bousaid, A., Theodoridis, T., and Nefti-Meziani, S. (2016). Introducing a novel marker-based geometry model in monocular vision. In *Positioning, Navigation and Communications (WPNC), 2016 13th Workshop on*, pages 1–6. IEEE.

[17] Brailov, A. Y. (2016). *Engineering Graphics: Theoretical Foundations of Engineering Geometry for Design*. Springer Publishing Company, Incorporated, 1st edition.

[18] Bresson, G., Alsayed, Z., Yu, L., and Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220.

[19] Bresson, G., Alsayed, Z., Yu, L., and Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220.

[20] Bretagne, E., Dassonvalle, P., and Caron, G. (2018). Spherical target-based calibration of terrestrial laser scanner intensity. application to colour information computation. *ISPRS journal of photogrammetry and remote sensing*, 144:14–27.

[21] Bräunl, T. (2008). *Localization and Navigation*, pages 241–269. Springer Berlin Heidelberg, Berlin, Heidelberg.

[22] Cao, Z. L., Oh, S. J., and Hall, E. L. (1986). Dynamic omnidirectional vision for mobile robots. *Journal of Robotic Systems*, 3(1):5–17.

[23] Caruso, D., Engel, J., and Cremers, D. (2015). Large-scale direct slam for omnidirectional cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148. IEEE.

[24] Caselitz, T., Steder, B., Ruhnke, M., and Burgard, W. (2016). Monocular camera localization in 3d lidar maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931. IEEE.

[25] Celozzi, C., Paravati, G., Sanna, A., and Lamberti, F. (2010). A 6-dof artag-based tracking system. *IEEE Transactions on Consumer Electronics*, 56(1):203–210.

[26] Chahine, G. and Pradalier, C. (2018). Survey of monocular slam algorithms in natural environments. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 345–352. IEEE.

[27] Chatila, R. and Laumond, J.-P. (1985). Position referencing and consistent world modeling for mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138–145. IEEE.

[28] Chavez-Garcia, R. and Aycard, O. (2015). Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10.

[29] Choi, S., Joung, J. H., Yu, W., and Cho, J. (2011). What does ground tell us? monocular visual odometry under planar motion constraint. In *2011 11th International Conference on Control, Automation and Systems*, pages 1480–1485.

[30] Corke, P. (2013). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Publishing Company, Incorporated, 1st edition.

[31] Davies, E. R. (2004). *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[32] Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Iccv*, volume 3, pages 1403–1410.

[33] DeMenthon, D. and Davis, L. S. (1992). Exact and approximate solutions of the perspective-three-point problem. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1100–1105.

[34] Desouza, G. N. and Kak, A. C. (2002). Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267.

[35] Diaz, J. C. F., Carter, W. E., Shrestha, R. L., and Glennie, C. L. (2017). *LiDAR Remote Sensing*, pages 929–980. Springer International Publishing, Cham.

[36] Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241.

[37] Ducard, G. and D'Andrea, R. (2009). Autonomous quadrotor flight using a vision system and accommodating frames misalignment. In *IEEE Fourth International Symposium on Industrial Embedded Systems, SIES 2009, Ecole Polytechnique Federale de Lausanne, Switzerland, July 8-10, 2009*, pages 261–264.

[38] Ducard, G. and D'Andrea, R. (2009). Autonomous quadrotor flight using a vision system and accommodating frames misalignment. In *2009 IEEE International Symposium on Industrial Embedded Systems*, pages 261–264.

[39] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110.

[40] DuToit, R. C., Hesch, J. A., Nerurkar, E. D., and Roumeliotis, S. I. (2017). Consistent map-based 3d localization on mobile devices. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6253–6260.

[41] Endres, F., Hess, J., Sturm, J., Cremers, D., and Burgard, W. (2014). 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187.

[42] Engel, J., Schöps, T., and Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer.

[43] Engel, J., Stückler, J., and Cremers, D. (2015). Large-scale direct slam with stereo cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942. IEEE.

[44] Fiala, M. (2005a). Artag, a fiducial marker system using digital techniques. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 590–596, Washington, DC, USA. IEEE Computer Society.

[45] Fiala, M. (2005b). Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE.

[46] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[47] Fleck, M. M. (1995). Perspective projection: the wrong imaging model. *Department of Computer Science, University of Iowa*, pages 1–27.

[48] Frost, D. P., Kähler, O., and Murray, D. W. (2016). Object-aware bundle adjustment for correcting monocular scale drift. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4770–4776.

[49] Frstner, W. and Wrobel, B. P. (2016). *Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction*. Springer Publishing Company, Incorporated, 1st edition.

[50] Fryer, J. G. and Brown, D. C. (1986). Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing*, 52(1):51–58.

[51] Gálvez-López, D. and Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

[52] Gao, X.-S., Hou, X.-R., Tang, J., and Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943.

[53] Gómez, C., Mattamala, M., Resink, T., and Ruiz-del-Solar, J. (2018). Visual slam-based localization and navigation for service robots: The pepper case. *CoRR*, abs/1811.08414.

[54] Haala, N., Peter, M., Kremer, J., and Hunter, G. (2008). Mobile lidar mapping for 3d point cloud collection in urban areas - a performance test. *In: Proceedings of XXI ISPRS Congress, Beijing, China, July 3-11, 2008*, 37.

[55] Haralick, B. M., Lee, C.-N., Ottenberg, K., and Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, 13(3):331–356.

[56] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition.

[57] He, D., Liu, X., Peng, X., Ding, Y., and Gao, B. Z. (2013). Eccentricity error identification and compensation for high-accuracy 3d optical measurement. *Measurement Science and Technology*, 24(7):075402.

[58] Heath, T. L. et al. (1956). *The thirteen books of Euclid's Elements*. Courier Corporation.

[59] Heath, T. L. and Euclid (1956). *The Thirteen Books of Euclid's Elements, Books 1 and 2*. Dover Publications, Inc., New York, NY, USA.

[60] How, J. P., Behihke, B., Frank, A., Dale, D., and Vian, J. (2008). Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 28(2):51–64.

[61] Hübner, P., Weinmann, M., and Wursthorn, S. (2018). Marker-based localization of the microsoft hololens in building models. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42(1).

[62] Hung, Y., Yeh, P.-S., and Harwood, D. (1985). Passive ranging to known planar point sets. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 80–85. IEEE.

[63] Hwang, S.-Y. and Song, J.-B. (2011). Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812.

[64] Hyyppä, J., Jaakkola, A., Chen, Y., and Kukko, A. (2013). Unconventional lidar mapping from air, terrestrial and mobile. In *Proceedings of the Photogrammetric Week*, pages 205–214. Wichmann/VDE Verlag Berlin, Germany.

[65] Kanellakis, C. and Nikolakopoulos, G. (2017). Survey on computer vision for uavs: Current developments and trends. *Journal of Intelligent & Robotic Systems*, 87(1):141–168.

[66] Kang, D.-J., Ha, J.-E., and Jeong, M.-H. (2006). A method for camera pose estimation from object of a known shape. In *Intelligent Computing in Signal Processing and Pattern Recognition*, pages 606–613. Springer.

[67] Kitt, B. M., Rehder, J., Chambers, A. D., Schonbein, M., Lategahn, H., and Singh, S. (2011). Monocular visual odometry using a planar road model to solve scale ambiguity.

[68] Kizil, U. and Tisor, L. (2011). Evaluation of rtk-gps and total station for applications in land surveying. *Journal of Earth System Science*, 120(2):215–221.

[69] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234.

[70] Kolski, S., Ferguson, D., Bellino, M., and Siegwart, R. (2006). Autonomous driving in structured and unstructured environments. In *2006 IEEE Intelligent Vehicles Symposium*, pages 558–563. IEEE.

[71] Kremer, J. (2007). Performance of the streetmapper mobile lidar mapping system in" real world". *Projects, Photogrammetric Week 07*, pages 215–225.

[72] Landau, H., Chen, X., Klose, S., Leandro, R., and Vollath, U. (2009). Trimble's rtk and dgps solutions in comparison with precise point positioning. In

Sideris, M. G., editor, *Observing our Changing Earth*, pages 709–718, Berlin, Heidelberg. Springer Berlin Heidelberg.

[73] Lee, B., Daniilidis, K., and Lee, D. D. (2015). Online self-supervised monocular visual odometry for ground vehicles. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5232–5238.

[74] Lee, B., Daniilidis, K., and Lee, D. D. (2015). Online self-supervised monocular visual odometry for ground vehicles. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5232–5238. IEEE.

[75] Lei, G., Zhitao, X., Jun, W., Fang, Z. L., and Jinghong, M. (2013). Monocular vision distance measurement method based on dynamic error compensation.

[76] Lemaire, T., Berger, C., Jung, I.-K., and Lacroix, S. (2007). Vision-based slam: Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364.

[77] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166.

[78] Liu, H., Chen, M., Zhang, G., Bao, H., and Bao, Y. (2018). Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[79] Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080.

[80] Lu, X. X. (2018). A review of solutions for perspective-n-point problem in camera pose estimation. In *Journal of Physics: Conference Series*, volume 1087, page 052009. IOP Publishing.

[81] Luhmann, T. (2014). Eccentricity in images of circular and spherical targets and its impact to 3d object reconstruction. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 45.

[82] Ma, X., Djouadi, S. M., Crilly, P., and Smith, S. (2013). Retracted: Navigation strategy with the aid of the theater positioning system. *Intelligent Control and Automation*, 4(02):206.

[83] Maganto, A. L., Menendez, J. M., Salgado, L., Rendon, E., and Garcia, N. (2000). A monocular vision system for autonomous vehicle guidance. In *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, volume 1, pages 236–239 vol.1.

[84] Marchand, E., Uchiyama, H., and Spindler, F. (2016). Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651.

[85] Marrón, M., Sotelo, M. Á., García, J. C., Fernández, D., and Parra, I. (2006). 3d-visual detection of multiple objects and structural features in complex and dynamic indoor environments. In *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pages 3373–3378. IEEE.

[86] Mather, G. (2016). *Foundations of sensation and perception*. Psychology Press.

[87] Mei, C., Sibley, G., Cummins, M., Newman, P. M., and Reid, I. D. (2011). RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94(2):198–214.

[88] Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P., and Savatier, X. (2017). A study of vicon system positioning performance. *Sensors*, 17(7):1591.

[89] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.

[90] Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

[91] Mutka, A., Miklic, D., Draganjac, I., and Bogdan, S. (2008). A low cost vision based localization system using fiducial markers. *IFAC Proceedings Volumes*, 41(2):9528–9533.

[92] N Malamas, E., Petrakis, E., Zervakis, M., Petit, L., and Legat, J.-D. (2003a). A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21:171–188.

[93] N Malamas, E., Petrakis, E., Zervakis, M., Petit, L., and Legat, J.-D. (2003b). A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, 21:171–188.

[94] Neale, W. T., Hessel, D., and Terpstra, T. (2011). Photogrammetric measurement error associated with lens distortion. Technical report, SAE Technical Paper.

[95] Niculescu, D. and Badri Nath (2003). Ad hoc positioning system (aps) using aoa. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1734–1743 vol.3.

[96] O' Riordan, A., Newe, T., Toal, D., and Dooly, G. (2018). Stereo vision sensing: Review of existing systems.

[97] Pire, T., Fischer, T., Civera, J., De Cristóforis, P., and Berlles, J. J. (2015). Stereo parallel tracking and mapping for robot localization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1373–1378.

[98] Prince, S. J. D. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, New York, NY, USA, 1st edition.

[99] Psiaki, M. L. (2006). Global navigation satellite systems: genesis, state of the art, and future directions. In *Plenary Presentation at the 46th Israel Annual Conference on Aerospace Sciences, Tel Aviv & Haifa, Israel*.

[100] Quan, L. and Lan, Z. (1999). Linear n-point camera pose determination. *IEEE Transactions on pattern analysis and machine intelligence*, 21(8):774–780.

[101] Rice, A. C., Harle, R. K., and Beresford, A. R. (2006). Analysing fundamental properties of marker-based vision system designs.

[102] Richter-Gebert, J. (2011). *Perspectives on Projective Geometry: A Guided Tour Through Real and Complex Geometry*. Springer Publishing Company, Incorporated, 1st edition.

[103] Riegler, G., Liao, Y., Donne, S., Koltun, V., and Geiger, A. (2019). Connecting the dots: Learning representations for active monocular depth estimation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[104] Rodehorst, V., Heinrichs, M., and Hellwich, O. (2008). Evaluation of relative pose estimation methods for multi-camera setups. *International Archives of Photogrammetry and Remote Sensing (ISPRS'08)*, pages 135–140.

[105] Royer, E., Lhuillier, M., Dhome, M., and Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260.

[106] Ruan, M. and Huber, D. (2014). Calibration of 3d sensors using a spherical target. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 187–193. IEEE.

[107] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011a). Orb: An efficient alternative to sift or surf. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA. IEEE Computer Society.

[108] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011b). Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer.

[109] Saito, S., Hiyama, A., Tanikawa, T., and Hirose, M. (2007). Indoor marker-based localization using coded seamless pattern for interior decoration. In *2007 IEEE Virtual Reality Conference*, pages 67–74. IEEE.

[110] Sakpere, W., Adeyeye-Oshin, M., and Mlitwa, N. B. (2017). A state-of-the-art survey of indoor positioning and navigation systems and technologies. *South African Computer Journal*, 29(3):145–197.

[111] Sasaki, T. (2011). Geometry , projective geometry , and non-euclidean geometry.

[112] Saxena, A., Chung, S. H., and Ng, A. Y. (2005). Learning depth from single monocular images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 1161–1168, Cambridge, MA, USA. MIT Press.

[113] Schechner, Y. Y. and Kiryati, N. (1998). Depth from defocus vs. stereo: how different really are they? In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, volume 2, pages 1784–1786 vol.2.

[114] Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The international Journal of robotics Research*, 21(8):735–758.

[115] Se, S., Lowe, D. G., and Little, J. J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on robotics*, 21(3):364–375.

[116] Se, S., Lowe, D. G., and Little, J. J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375.

[117] Segvic, S., Remazeilles, A., Diosi, A., and Chaumette, F. (2007). Large scale vision-based navigation without an accurate global reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[118] Shih, T. and Faig, W. (1988). A solution for space resection in closed form. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 27:547–556.

[119] Shiozaki, T. and Dissanayake, G. (2018). Eliminating scale drift in monocular slam using depth from defocus. *IEEE Robotics and Automation Letters*, 3(1):581–587.

[120] Singh, S., Simmons, R., Smith, T., Stentz, A., Verma, V., Yahja, A., and Schwehr, K. (2000). Recent progress in local and global traversability for planetary rovers. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1194–1200 vol.2.

[121] Skog, I. and HÃ¤ndel, P. (2019). State-of-the art and future in-car navigation systemsâ€“a survey.

[122] Smith, R. C. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68.

[123] Song, S., Chandraker, M., and Guest, C. C. (2016). High accuracy monocular sfm and scale correction for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):730–743.

[124] Song, Z., Jiang, G., and Huang, C. (2011). A survey on indoor positioning technologies. In Zhou, Q., editor, *Theoretical and Mathematical Foundations of Computer Science*, pages 198–206, Berlin, Heidelberg. Springer Berlin Heidelberg.

[125] Stentz, A. and Hebert, M. (1995). A complete navigation system for goal acquisition in unknown environments. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 425–432 vol.1.

[126] Strasdat, H., Montiel, J., and Davison, A. J. (2010). Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3):7.

[127] Sánchez, A. P., Andrade-Cetto, J., and Moreno-Noguer, F. (2013). Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2387–2400.

[128] T., D. (1999). Discovery of linear perspective and its limitations.

[129] Taketomi, T., Uchiyama, H., and Ikeda, S. (2017). Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16.

[130] Tang, Z., von Gioi, R. G., Monasse, P., and Morel, J.-M. (2017). A precision analysis of camera distortion models. *IEEE Transactions on Image Processing*, 26(6):2694–2704.

[131] Tappero, F. (2009). Low-cost optical-based indoor tracking device for detection and mitigation of nlos effects. *Procedia Chemistry*, 1(1):497–500.

[132] Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1023–1029 vol.2.

[133] Vass, G. and Perlaki, T. (2003). Applying and removing lens distortion in post production. In *Proceedings of the 2nd Hungarian Conference on Computer Graphics and Geometry*, pages 9–16.

[134] Walker, S. (2015). Close-range photogrammetry and 3d imaging. *Photogrammetric Engineering & Remote Sensing*, 81(4):273–274.

[135] Won, D. H., Lee, Y. J., Sung, S., and Kang, T. (2008). Integration of vision based slam and nonlinear filter for simple mobile robot navigation. In *2008 IEEE National Aerospace and Electronics Conference*, pages 373–378.

[136] Wu, Y., Tang, F., and Li, H. (2018). Image-based camera localization: an overview. *Visual Computing for Industry, Biomedicine, and Art*, 1(1):8.

[137] Yagi, Y. (1999). Omnidirectional sensing and its applications. *IEICE Transactions on Information and Systems*, 82(3):568–579.

[138] Yassin, M. and Rachid, E. (2015). A survey of positioning techniques and location based services in wireless networks. In *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, pages 1–5.

[139] Yin, X., Wang, X., Du, X., and Chen, Q. (2017). Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5870–5878.

[140] Yin, Y., Wang, M., Liu, X., and Peng, X. (2012). Center location of circular targets with surface fitting method. In *Applications of Digital Image Processing XXXV*, volume 8499, page 84991U. International Society for Optics and Photonics.

[141] Zafari, F., Gkelias, A., and Leung, K. (2017a). A survey of indoor localization systems and technologies. *arXiv preprint arXiv:1709.01015*.

[142] Zafari, F., Gkelias, A., and Leung, K. K. (2017b). A survey of indoor localization systems and technologies. *CoRR*, abs/1709.01015.

[143] Zhang, W. and Kosecka, J. (2006). Image based localization in urban environments. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 33–40.

[144] Zhichao Chen and Birchfield, S. T. (2006). Qualitative vision-based mobile robot navigation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2686–2692.